SynchroNet ES/2

Service Manual

527-194
Issue 1

As part of our policy of continual product improvement, we reserve the right to alter specifications without notice but with due regard to all current legislation.

HEAD OFFICE

AMS NEVE PLC • BILLINGTON ROAD • BURNLEY
LANCS BB11 5UB • ENGLAND
TELEPHONE: +44 (0) 1282 457011 • FAX: +44 (0) 1282 417282

LONDON OFFICE

TELEPHONE: +44 (0) 20 7916 2828 • FAX: +44 (0) 20 7916 2827

NORTH AMERICAN OFFICES

AMS NEVE INC., NEW YORK
TEL: +1 (212) 965 1400 • FAX: +1 (212) 965 3739
AMS NEVE INC., HOLLYWOOD
TEL: +1 (818) 753 8789 • FAX: +1 (818) 623 4839
RUPERT NEVE CANADA INC., TORONTO
TEL: +1 (416) 365 3363 • FAX: +1 (416) 365 1044
e-mail: enquiry@ams-neve.com
http://www.ams-neve.com

CIRCUIT DESCRIPTION

Introduction

The ES/2 synchroniser is a multi-processor machine controller capable of interfacing with ATR, VTR and film machines as well as several forms of controllers.

The ES/2 has two AK ESbus ports each of which contains an RS422 communications port. The ESbus ports carry

the Timeline - a system wide reference signal

a TLext line - to indicate presence or absence of a timeline signal

a frame bus - used to distribute a video sync locked signal

a mute bus - or system locked line.

The general purpose parallel machine port is capable of

issuing transport commands

reading tallies

reading tach

generating pulse information for film machines

generating a DC or FM servo control signal.

Two additional communication ports are

an RS422/232 machine serial control port allowing control of serial video machines

a diagnostics port allowing remote interrogation of the system.

The ES/2 also features

A flexible timecode reader and generator.

A reference input capable of decoding video syncs or square wave pulses.

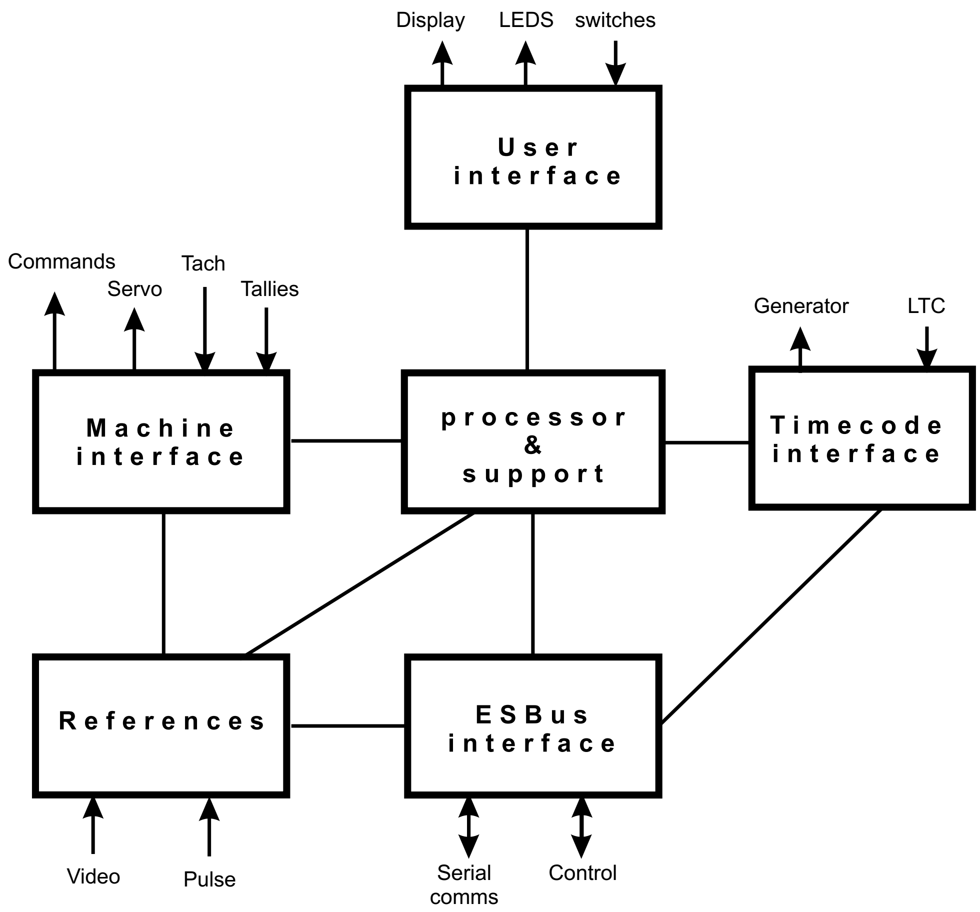

Figure 1 : ES/2 system block diagram

The ES/2 circuit can be split into the following main blocks and is shown in *Figure 1*.

Microprocessor.

The ES/2s Computing power required to control the machine and communicate to the outside world.

Machine interface.

Providing the means to control and synchronise a wide range of ATR, DTR, VTR and film machines.

Timecode interface.

Conditioning for positional reference.

User interface.

Human input and feedback via keys, buttons, LEDs and the LCD display.

ESbus interface.

Providing control signals and the means to interface with an ESbus controller, or other.

References.

Allowing a range of system reference options.

A large part of the ES/2s digital circuitry is contained within four Field Programmable Gate Arrays (FPGAs). These FPGAs have been named MCN, SUPPORT, SPLIT and DECODE. The FPGAs are shown on the block diagrams as a dotted box with the associated FPGA's name.

MICROPROCESSOR AND SUPPORT

There are two Motorola M6809E processing units within the ES/2. The KERNEL processor controls the machine and provides the user interface. The COMMS processor deals exclusively with the AK-ESbus interface, including ESbus and emulation modes.

The processors require the following conditions to run :

Low pulse on the reset line.

Halt line held high

Clock signal ‘E’

Clock signal ‘Q’ leading ‘E’ by 90°.

EPROM situated at the top of address space.

RAM available.

The processor part of the ES/2 circuit, shown in *Figure 2*, can be split into the following parts:

Power monitor.

Clock generator.

Kernel processor, RAM and ROM.

Comms processor, RAM and ROM and CTC.

Address decoding.

Kernel / Comms interface

Diagnostics port

Firq generator.

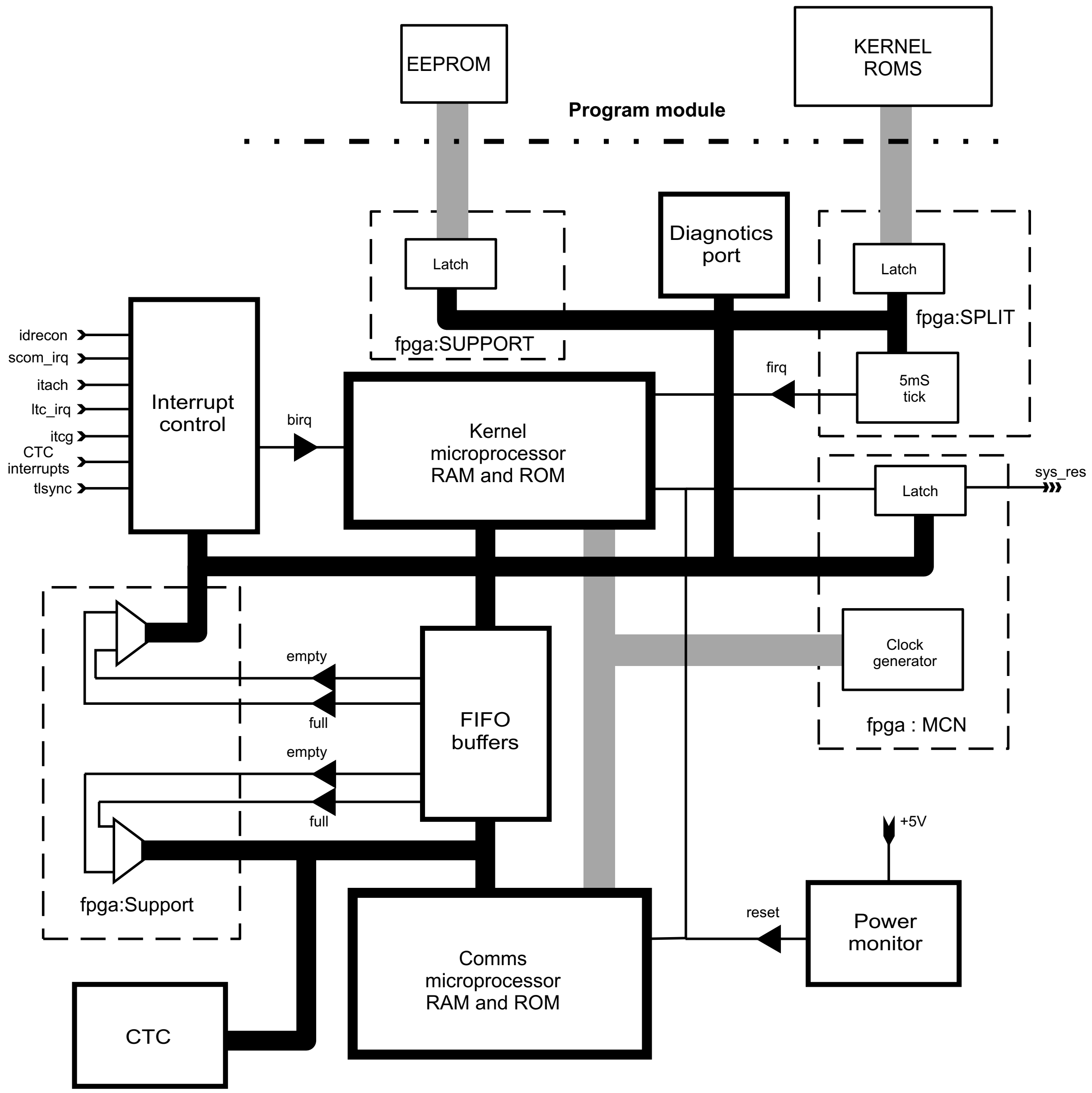

Figure 2 : Processor and support

Power monitor

The power monitoring is contained in U50 (sheet 6), a MAX 791 microprocessor supervisor chip. This chip provides the ES/2 with the following :

System reset line, !RES

A power fail signal, !NMI

Chip select protection for the battery backed up RAM during power down.

RAM supply switching between the main +5V supply and the battery.

The reset line !RES resets both of the microprocessors. The peripherals are reset by a signal called !SYS_RES. This signal is generated by the FPGA MCN and is always active when !RES is active. A software controlled reset can also be sent to all peripherals via this line.

The MAX791 continuously monitors its +5v supply for power supply failure. When the +5v supply falls below 4.8V then the !NMI line is pulled low. The reset is active when the +5V supply is below 4.65V. During power up when the supply crosses the 4.65V threshold the !RES line stays active for 200ms.

The MAX791 also looks after the supply and chip select enable signal to the battery backed up RAM. When the +5V supply is above 4.65V the RAM (BVCC line in the ES/2) is supplied from the main +5V line. If the +5V supply falls below this threshold then the RAM is supplied from the battery and the chip select line (!BBRAM) is disabled. On powering up the MAX791 will disable the second chip select pulse if the battery voltage is below 2V. If this is the case then the microprocessor recognises the missing access and displays the battery voltage low warning message.

Clock generator

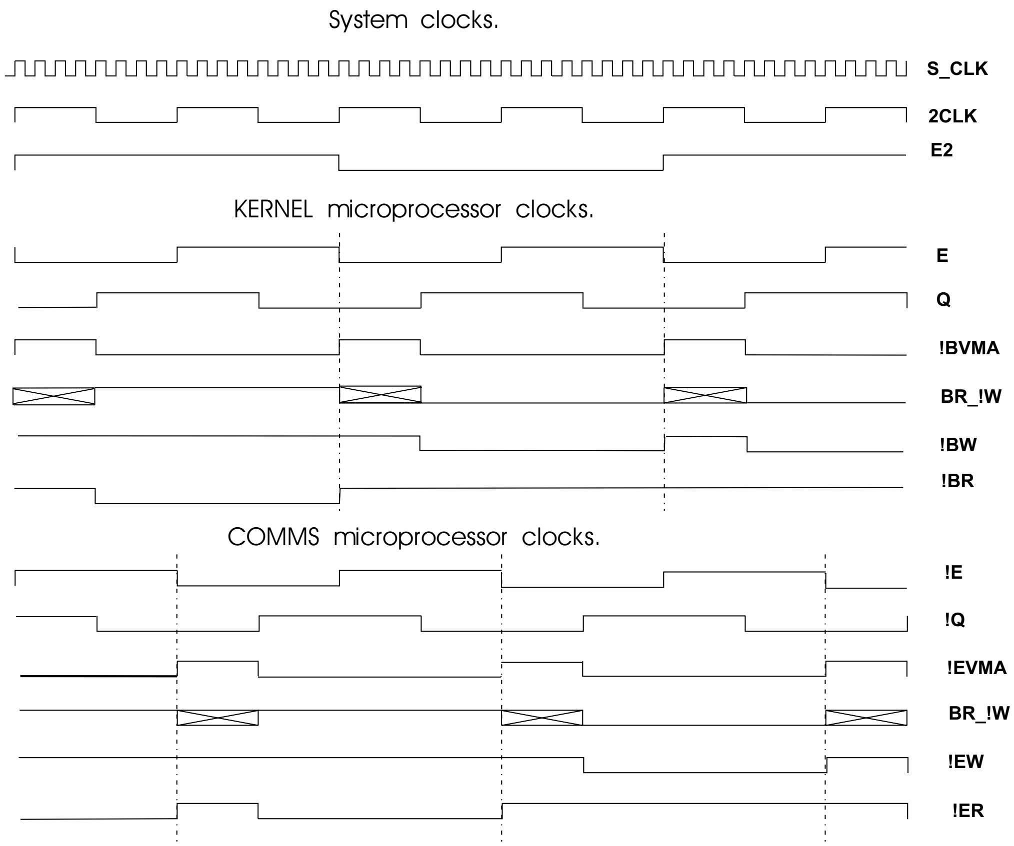

Figure 3 : ES/2 clock signals

All system clock signals are derived from the main 16MHz clock (S_CLK) which is generated by X1. The KERNEL and COMMS microprocessors are running on alternate phases. Other than reducing clock skew there is no technical reason for this, the processors operate completely asynchronously. *Figure 3* shows the relationship between clock signals and a description of the purpose of each clock signal follows.

CLK	4MHz clock. Used by the ESbus and machine SIO chips - U44 and U49
E2	1MHz clock. Used as timer references for the KERNEL CTCs - U44, U45, U46.
E	2MHz clock. The main KERNEL microprocessor clock - U48.
Q	2MHz clock. The KERNEL microprocessor address strobe signal.
!BVMA :	2MHz clock. The KERNEL microprocessor's valid memory address signal. It is used by the FPGA DECODE as a gating signal for certain peripheral chip select lines.
BR/!W :	Signal generated by the KERNEL microprocessor to allow peripherals to distinguish between a read and write cycle.
!BW	This signal is active low during a KERNEL write cycle. It is the BR/!W signal gated with !BVMA and is used by the SIO chips U49.
!BR	This signal is active low during a KERNEL read cycle. It is the BR/!W signal gated with !BVMA and is used by the SIO chip U49 and the ROMs.
!E	2MHz clock. The main COMMS microprocessor clock - U44.
!Q	2MHz clock. The COMMS microprocessor address strobe signal.
!EVMA	2MHz clock. The COMMS microprocessor's valid memory address signal. Used by the FPGA DECODE to gate certain peripheral chip select signals.
ER/!W	Signal generated by the COMMS microprocessor to allow peripherals distinguish between a read and write cycle.
!EW	This signal is active low during a COMMS write cycle. It is the BR/!W signal gated with !BVMA and is used by the SIO chip U44.
!ER	This signal is active low during a COMMS read cycle. It is the BR/!W signal gated with !BVMA and is used by the SIO chip U44 and the COMMS ROM on the program module.

KERNEL Processor, RAM and ROM

The KERNEL microprocessor (U48) is responsible for the following:

User interface.

Machine control.

Controlling ESbus signal routine.

Switching and measuring reference signals.

Controlling the timecode generator.

KERNEL and MACHINE ROMs

The KERNEL's main ROM - U2 on the program module - is a 27C1001 (or 27C010 depending upon the manufacturer).

A 27C1001 is a 1Mbit or 128Kbyte ROM but the 6809 can only address a maximum of 64K directly and so the ROM is arranged into 8 pages of 16Kbytes. The ROM's address select lines A0 - A13 are generated by the KERNEL microprocessor while the top 3 address lines are generated by a latch within the FPGA SPLIT. This latch decodes the ROM's chip select line (!KERN), the KERNEL's data lines (BD0 - BD2) and the R/!W line (!BR/!W) and outputs the lines KA_14, KA_15 and KA_16 which are used to select the current page of the ROM. When the KERNEL's program wishes to change page it simply writes the page number to a location in the ROM space.

The ES/2's MACHINE ROM is also a paged ROM, currently a 27C512 although there is the capability for fitting 27C1001s in future. The page selection of the MACHINE ROM is similar to the KERNEL except that the top address line is shared with the KERNEL ROM (KA16).

KERNEL RAM

The KERNEL microprocessor's RAM (U53 - sheet 1) is a battery backed up 62256 which is a 32Kbyte RAM. The RAM occupies only 16k of address space and so 50% of the RAM is unused.

A MAX791 (U50 - sheet 6) provides supervision of the battery backup. It automatically switched the RAM's supply (BVCC) between the main supply and the battery when the +5V supply falls below 4.65V. It also disables the chip select line (!BBRAM) at this point to prevent spurious writes to any RAM location.

KERNEL EEPROM

To allow the user to select hard reset defaults there is a 512 byte EEPROM on the program module (U1). The EEPROM used is a National Semiconductors 24C04 (NOTE : some manufacturers have different pin outs for this chip) which contains an I2C interface. The FPGA SUPPORT generates the two control lines : a clock signal (EE_C) and a bi-directional data line (EE_D). These signals are only active when the microprocessor is communicating with the EEPROM.

COMMS processor, RAM, ROM and CTC.

The COMMS microprocessor is solely responsible for dealing with the protocol translation between the ESbus ports and the KERNEL microprocessor. Currently the ES/2 COMMS can communicate using AK ESbus dialect, SONY P2 protocol, Zeta 3 and 2600 protocols.

COMMS ROM - U4 on the program module - is a 27C256 occupying 32Kbytes in the address space. The COMMS RAM is a non backed up 6264 - U42 on the circuit diagrams.

The COMMS CTC is a 68B40 timer counter chip - U43 on the circuit diagrams - and is used purely for event timing. There are 3 timers inside a 68B40 and the COMMS CTC provides the following functions:

1. Timer 1 interrupts every 10ms and is used for:
    a. Flashing the COMMS LED in emulate mode.
    b. Chase master message scheduling and chase master conflict timing.
2. Timer 2 is used to measure the protocol's inter-character timeout and break processing.
3. Timer 3 is used to time the regular ESbus cyclic reports.

Address Decode

All of the microprocessor address decoding is contained within the FPGA DECODE (U37) and can be split between the COMMS and KERNEL processor's decode.

COMMS decoding

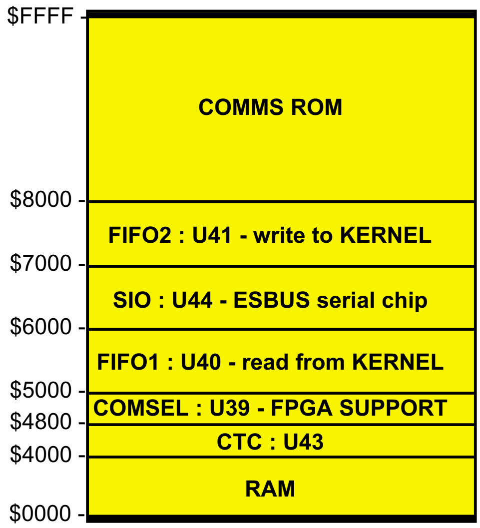

Figure 4 : COMMS processor memory map.

The COMMS chip select lines are generated from the address lines EA11 through to EA15 and gated with the !EVMA enable signal. *Figure 3* shows this in the form of an address map.

For more information see the information on FPGA decode under Kernel and Machine ROMs above.

The COMMS chip select output signals are:

!EROM - 32K	U4 on the program module, a 27C256 EPROM.
!ERAM - 8K	U42, a 6264 non battery backed up RAM.
!SIO - 4 locations	U44, a Z84C42 serial interface chip.
!ECTC - 8 locations	U43, a 68B40 counter timer chip.
!EFIFO1 - 1 locations	U41, read from the KERNEL interface.
!EFIFO2 - 1 locations	U42, write to the KERNEL interface.
!COMSEL - 2 locations	U39, read FIFO flags and write to comms LED latch

KERNEL decoding

The KERNEL chip select lines are generated from the address lines BA7 through to BA15 and are either gated with the !BVMA or the E signal, depending on the timing requirement of the device being selected. *Figure 4* shows this in the form of an address map. For more information see the information on FPGA DECODE in chapter 2. The KERNEL chip select output signals are :

!ROM - 8x16K	U2 on the program module, a 27C1001 EPROM.
!MACH - 8x16K	U3 on the program module, a 27C1001 EPROM.
!BBRAM - 16K	U53, a 62256 battery backed up RAM - 50% used.
!SCOM - 4 locations	U49, a Z84C42 serial interface chip.
!CTC1 - 8 locations	U54, a 68B40 counter timer chip.
!CTC2 - 8 locations	U55, a 68B40 counter timer chip.
!CTC3 - 8 locations	U56, a 68B40 counter timer chip.
!BFIFO1 - 1 location	U42, read from the COMMS interface.
!BFIFO2 - 1 location	U41, write to the COMMS interface.
!ICU1 - 2 locations	Master interrupt control unit.
!ICU2 - 2 locations	Slave interrupt control unit.
!INTA - 1 location	IRQ source clear.
!VITC - 16 locations	VITC expansion slot.
!LTCR - 8 locations	U37, LTC reader chip.
!BHOLE - 256 locations	BLACK HOLE expansion slot.
!TALLY - 1 location	U21, Tally buffer.
!CMD - 1 location	U25, Command latch
!LED - 1 location	U31, LED latch
!DAC - 2 locations	U9, DC servo DAC.
!SUPPORT - 10 locations	U39, control ports.
!SPLIT - 6 locations	U38, control ports.
!MCN - 4 locations	U36, control ports.

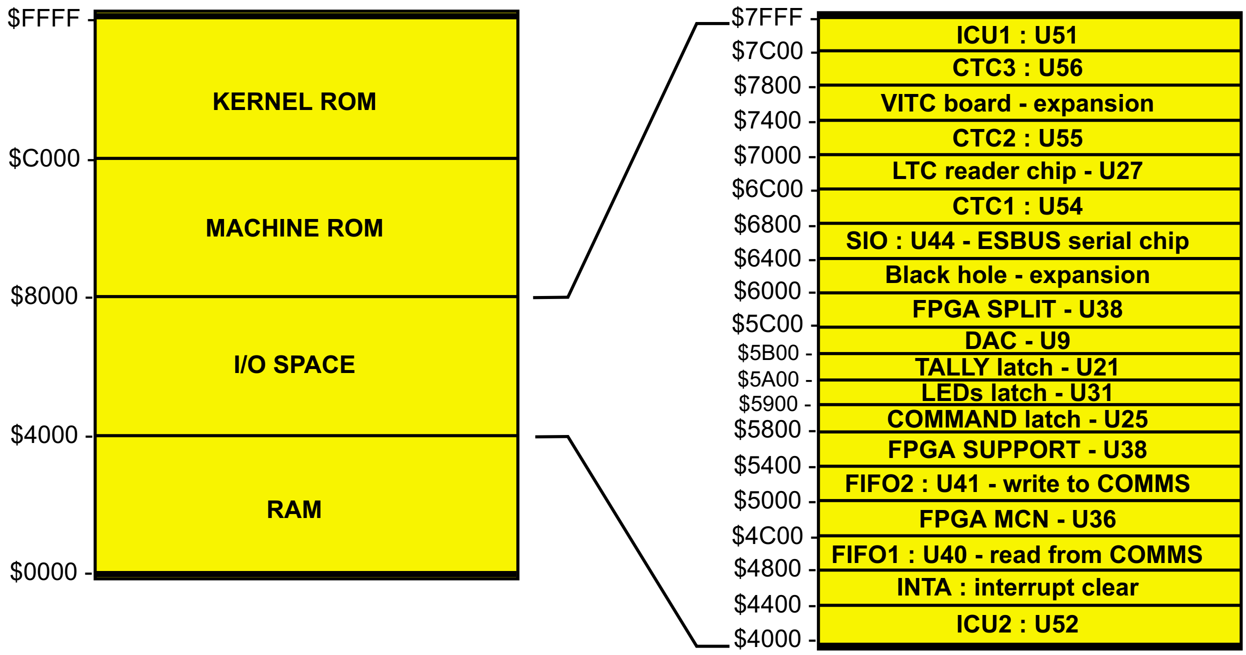

Figure 5 : KERNEL processor memory map.

Diagnostic port

The ES/2 diagnostic port is a general purpose RS232 serial port which allows the ES/2 to communicate with a PC. Half of the SIO U49 is used for the diagnostics (the other half for serial machine control). A 75155 RS232 transceiver is used to convert TTL levels to RS232. The diagnostic port communicates at 9600 baud with no parity, 8 data bits and 1 stop bit. A list of commands will be displayed when a ‘?’ followed by return is received.

Kernel / Comms interface

The KERNEL and COMMS microprocessors communicate with each other by two uni-directional First-In-First-Out buffers (FIFOs). The circuit diagrams show 2 MK4501 FIFOs. FIFO1 (U40) routes messages from the COMMS to the KERNEL while FIFO2 (U41) is concerned with sending messages from the KERNEL to the COMMS.

There are two hardware status lines generated by each FIFO which inform the processors when each FIFO is either full or empty. Both processors read these lines via buffers inside the FPGA SUPPORT. The KERNEL and COMMS processors have the same software to read and transmit to the FIFOs, a state diagram is shown in *Figure 6*.

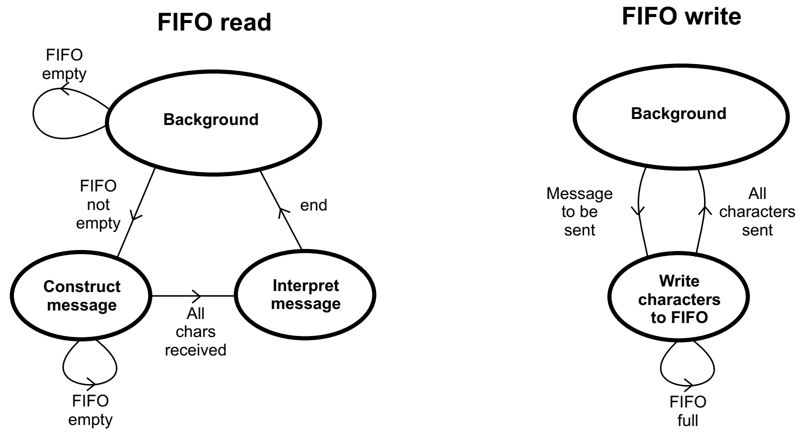

Figure 6 : FIFO software diagram.

FIRQ generator

The FIRQ line on a 6809 microprocessor is the second of three interrupt lines, NMI and IRQ being the other two. It is used in the ES/2 for scheduling various software routines including :

Polling the keys.

Tach timeout.

Diagnostic port processing.

Checking video periods.

FPGA SPLIT (U38) contains the firq generator. It simply counts the main system ‘Q pulse’ and generates a low going pulse on the FIRQ line after 5ms. The firq subroutine then cancels the low level by writing to a location which clears the output latch.

MACHINE INTERFACE.

The ES/2 machine interface is capable of controlling a wide range of audio, video and film machines. The circuitry for which can be split into the following sections:

Commands and tallies.

Tach interface.

Servo.

Serial machine interface.

Sony 5000 shuttle control.

Commands and Tallies

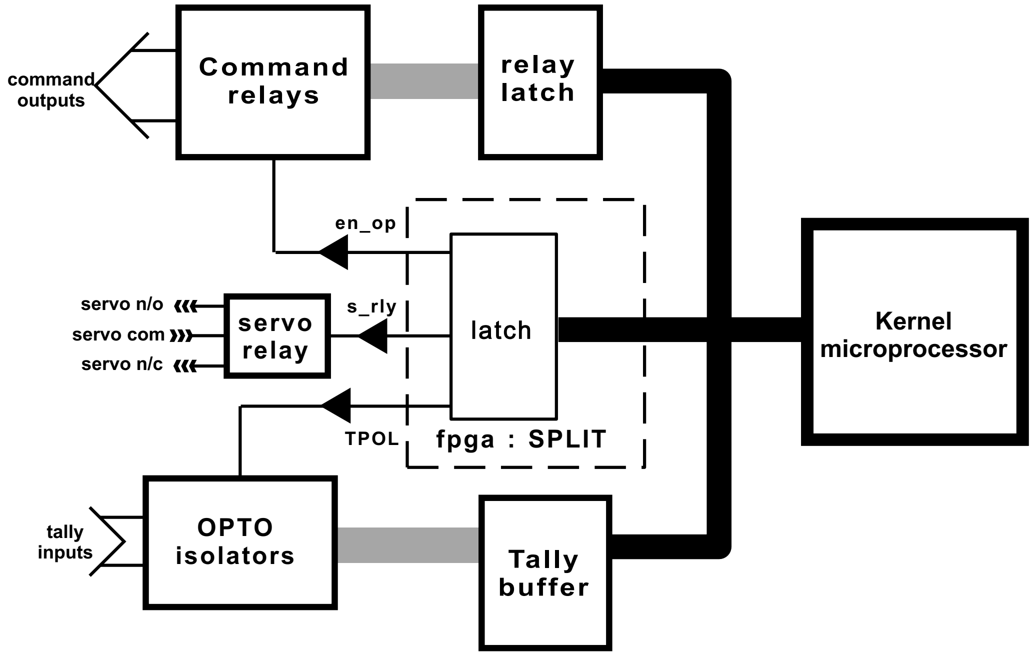

Figure 7 : Command output and tally input.

Figure 7 shows a block diagram of the command and tally interface.

There are 8 command relays used for transport commands : PLAY, STOP, RECORD, FORWARD WIND, REWIND, PAUSE, REHEARSE and LIFTER DEFEAT. Each relay has a common reference (CMDREF) which goes to the parallel machine connector to allow each interface to connect the reference to a level suitable for the machine to be controlled. An independent change-over relay is provided for switching servo enable signals.

The KERNEL microprocessor controls the relays via a 74HC377 octal latch (U25), the outputs of which drive the relays through an open collector driver. The relay outputs are disabled during power up by Q2. The microprocessor enables them by setting the EN_OP line low. The servo relay (K7) is separate from the command relays and is driven by the S_RLY line generated by the FPGA SPLIT.

The ES/2 is capable of reading 4 tallies from the machine : RECORD, End of Tape, RECORD ENABLE and spare. Each tally input is fed into 2 opto-isolators whose common go to the parallel machine connector to allow the ES/2 to read tallies of either sense. The outputs of the optos can be read by the KERNEL microprocessor via the tally buffer - U21, a 74HC374 - also used to read the

tach direction and the machine cable resistors. TPOL is used to invert the sense of the tallies so that the microprocessor always reads high for an active tally.

Tach interface

The ES/2 can read bi-phase or tach and direction across a wide range of speeds. Each time the ES/2 is calibrated the tach rate is compared with the timecode and a tach divider is set up - unless disabled by the machine file (top bit of tchdiv set) - to receive a tach rate as close to 1 pulse per frame as possible. *Figure 8* shows a block diagram of the tach circuitry. The circuit diagrams show the tach analogue conditioning, the tach phase locked loop and the FPGA part of the tach circuit.

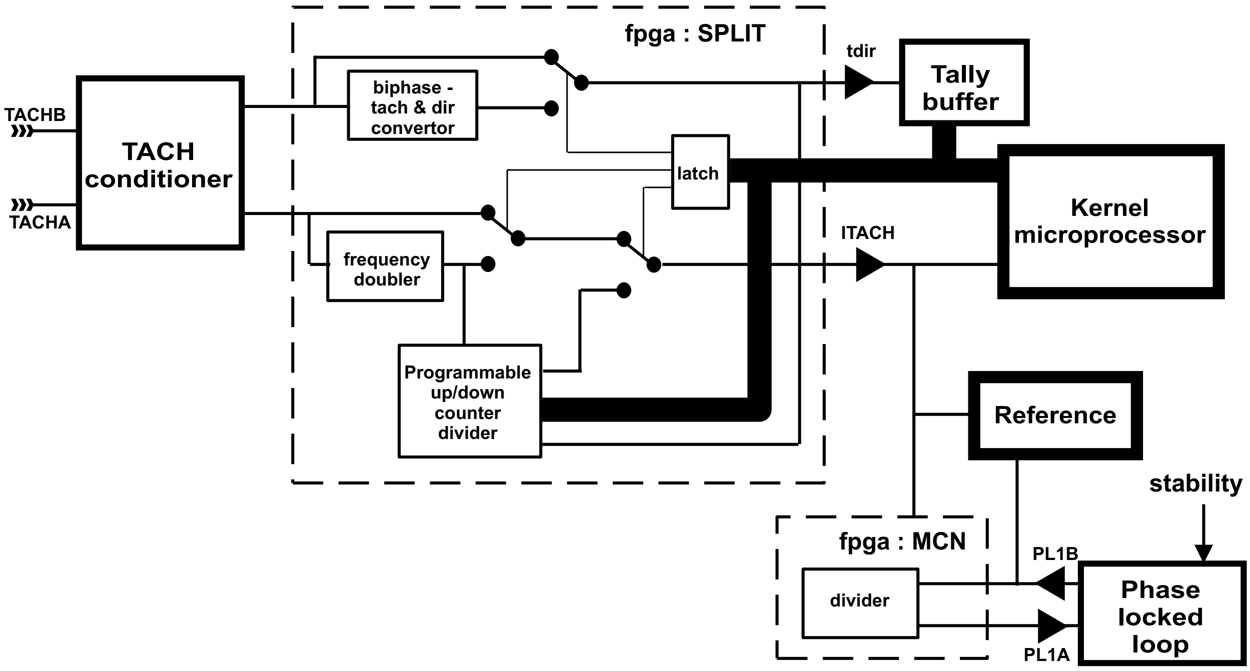

Figure 8 : Tach interface circuit.

There are 3 tach inputs, TACHA, TACHB and TACH COM. The TACH COM line goes to the parallel machine port to allow each interface to be wired to a potential suitable to the machine. The TACHA and TACHB signals are buffered by U28 and fed into U26 - a 4583. U26 is a schmitt trigger with RI4B and RI4C selecting the hysteresis. The output signals TACH1 and TACH2 are fed directly into FPGA SPLIT.

Inside the FPGA SPLIT there are three sections.

> The KERNEL can select the direction signal (TDIR) between the TACH2 signal (tach and direction) or the output of a bi-phase to tach and direction converter (bi-phase tach).
>
> There is a frequency doubler to increase the low frequency range.
>
> There is a 5 bit divider counter to enable the ES/2 to divide down high frequency tach rates by up to 32 times.

There are 2 tach outputs from the FPGA SPLIT. A direction signal (TDIR) which is read by the tally buffer U21 and the tach signal !ITACH. !ITACH is used for the following

A reference for the 5 x tach phase locked loop. The 5 x tach signal is used for locking a tach locked machine to an equivalent 5 x system reference, TL5.

An input to the KERNEL CTC to measure the period of the tach.

The tach interrupt to the KERNEL microprocessor. !ITACH is a active low latched pulse which is cleared by the tach interrupt.

Servo

The servo section of the ES/2 hardware has several different functions:

Provides an FM or DC signal to be used for servo locking or shuttling the machine.

Generates bi-phase or tach & direction signals to control a film machine.

Provides the necessary signals to read the machine cable identification resistors.

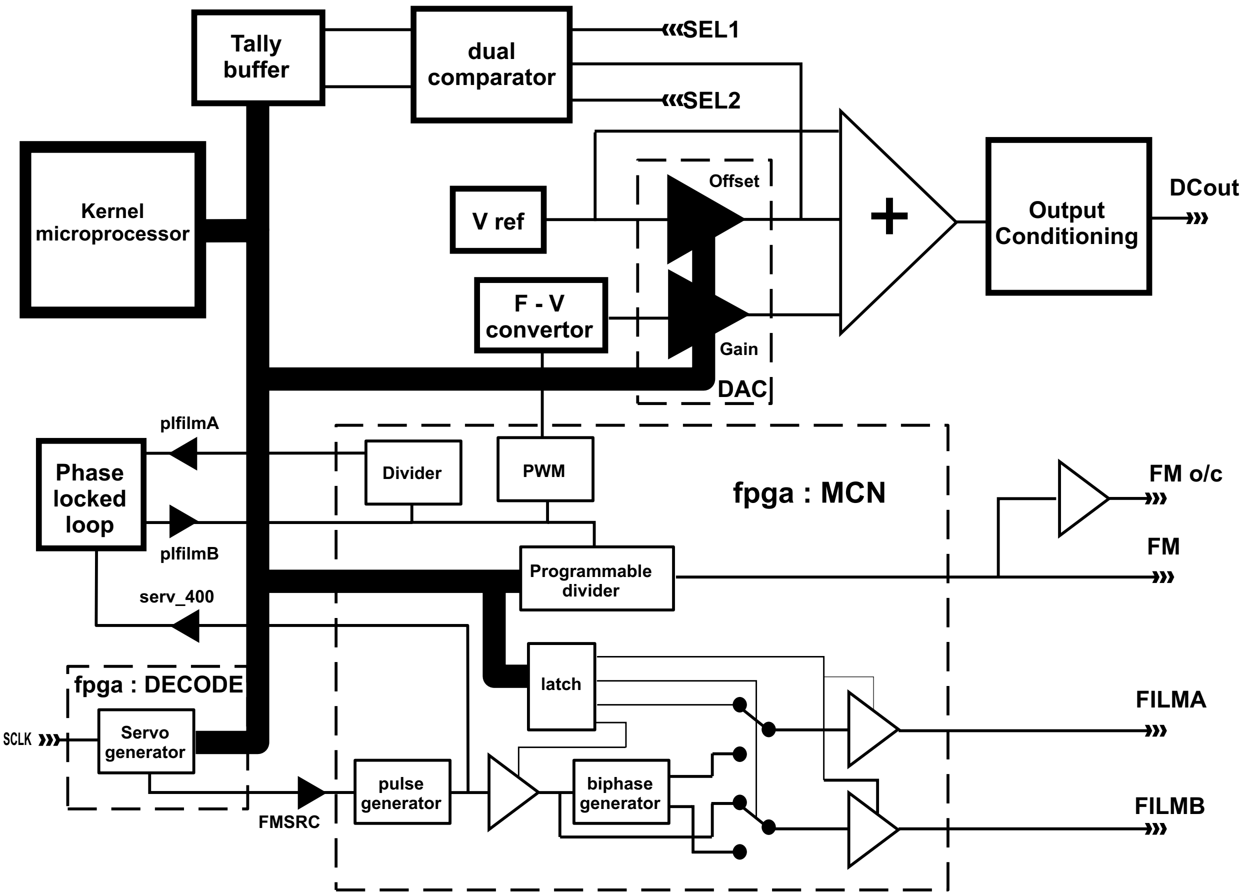

Figure 9 : Servo

A block diagram of the servo circuit is shown in *Figure 9*.

The Kernel controls the basic frequency of the servo by pre-setting a 24 bit servo generator inside the FPGA DECODE. FMSRC is the output and will be 200Hz when the generator is set to 40000 or 9C4016. FMSRC is fed into the FPGA MCN where it is converted into 400Hz pulses. The servo path then splits into two parts :Film machine and DC/FM servo.

1. Film machine

The film machine path goes through an enable gate and into a switch which selects between tach and direction and bi-phase. The signals are then fed to an output buffer which can be enabled/disabled.

2. DC/FM servo

For the FM/DC path the 400Hz pulses are fed into the servo phase locked loop whose output plfilmB is 89 times it's input - serv_400. This is fed back into FPGA MCN where it again splits into two : FM path and DC path.

2a. FM Servo

For the FM servo the PLFILMB signal is fed into a programmable divider which is controlled by software. The parameter svodiv is used to program this divider. The output of the divider is the FM signal which goes to the parallel machine control port. An open collector FM signal is also provided via U23.

2b. DC Servo

The DC servo is a little more intricate. The PLFILMB signal is converted to a pulse width modulated - PWM - signal which is the DC output of the FPGA MCN. For a nominal servo generator value of 40000 or 9C4016 the mark space ratio of the signal DC is approximately 50 - 50. When the servo frequency is decreased the mark space ratio also decreases. This signal can be inverted by software control for machines which require the opposite polarity.

The PWM DC signal is converted to a DC voltage by U13, a frequency to voltage converter shown on sheet 5. The relationship between the output of the V-F converter and the servo frequency is shown in figure 10. The precise shape of this curve is dependent on the resistor, capacitor and Vcc tolerances of an individual ES/2 and so *Figure 10* should only be taken as a guide.

The output of the V-F converter is fed into the reference of a D/A converter (half of U9 - a dual D/A converter). This part of the D/A has the effect of multiplying the reference signal by the value in the D/A converter. This value is the GAIN parameter from the machine file.

The second half of the D/A converter is set to the value of the parameter OFFSET from the machine file. It is used to adjust the zero point of the DC swing.

Both of the D/A outputs are fed into a summing amplifier - U2 - along with a -10V reference which has the effect of inverting the sense of the output.

The output of this amplifier drives the DCout line which goes to the parallel machine connector.

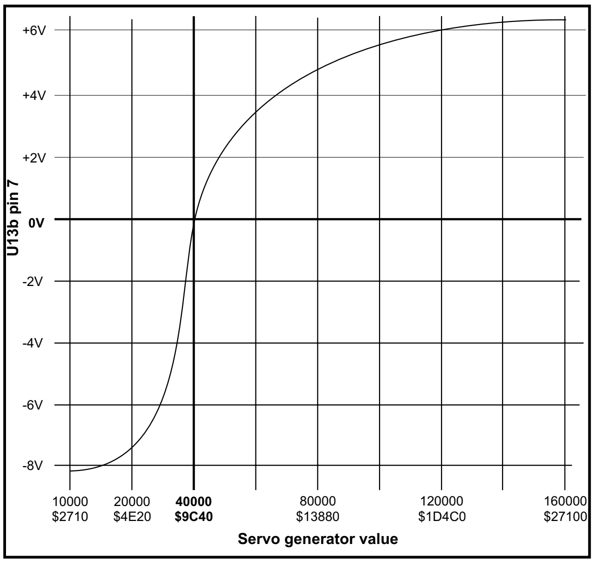

Figure 10 : Graph of Vf vs servo generator value

The DCout voltage follows the characteristic shown in the equation :

$$DCout \quad \frac{20}{256} * OFFSET \quad \frac{GAIN}{256} * (9.75 \quad 3.9 * Vf) \quad 10$$

Where :

DCout	=	DC output - TP5.
Vf	=	V-F converter output.
GAIN	=	Parameter 100A16.
OFFSET	=	Parameter 100B16.

Machine selector resistors

The DC servo is also used to read the machine selector resistors on the interface cable. There are 2 resistors on each cable which form part of a potential divider. The output of this potential divider is fed into a dual comparator (U1 on sheet 5) which compares it's input with the output from the U9's OFFSET DAC. The KERNEL microprocessor is able to monitor the outputs of the comparator (MS1 and MS2) via the TALLY buffer U21 while it increases the value in the OFFSET DAC. When the MS1 and MS2 lines change state the microprocessor is able to select the correct machine file from the machine ROM.

Serial machine control

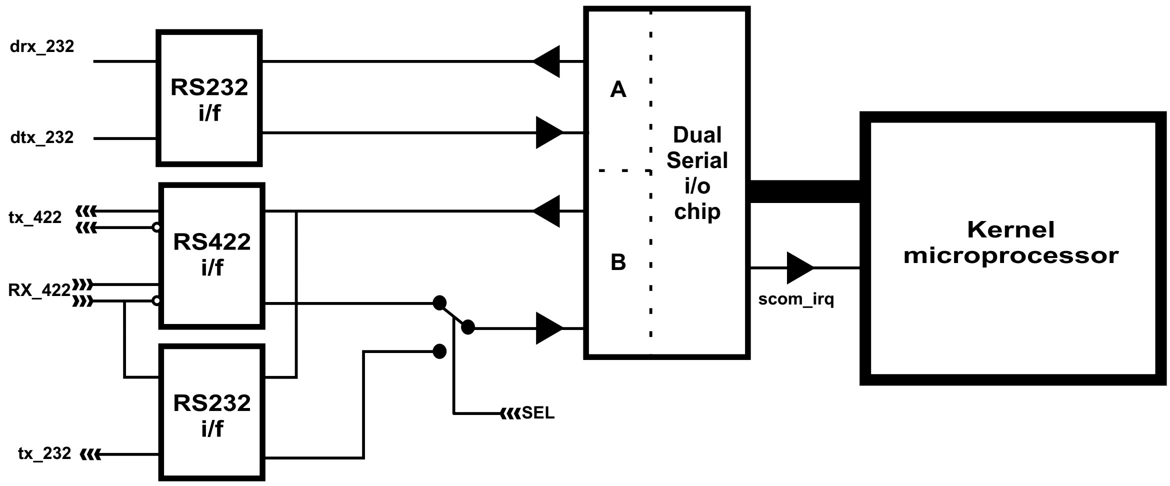

Figure 11 : Serial machine control

The ES/2 KERNEL microprocessor has a dual serial interface chip (U44). One side of this chip is used for the diagnostics port while the other is used to control serial machines. The port is capable of communicating with either RS232 or RS422 standards depending on the SEL line on the serial machine connector port. This line selects which receive line is decoded by the serial chip. For RS232 the SEL line is tied to 0V and for RS422 the SEL line should be left open. *Figure 11* shows a block diagram of the serial port.

Sony serial.

Sony 5000 series video machine, such as the quite popular 5850, require a serial stream to control shuttle mode. The machine outputs a clock signal with a sync waveform every 16 bits. The ES/2 buffers the clock waveform with U22F and converts the sync square wave to a high going pulse every rising edge. The ES/2 outputs a high going edge during the bit number 13 to command the machine to enter shuttle mode. No other commands are used by the ES/2. *Figure 12* shows a block diagram of the Sony serial interface.

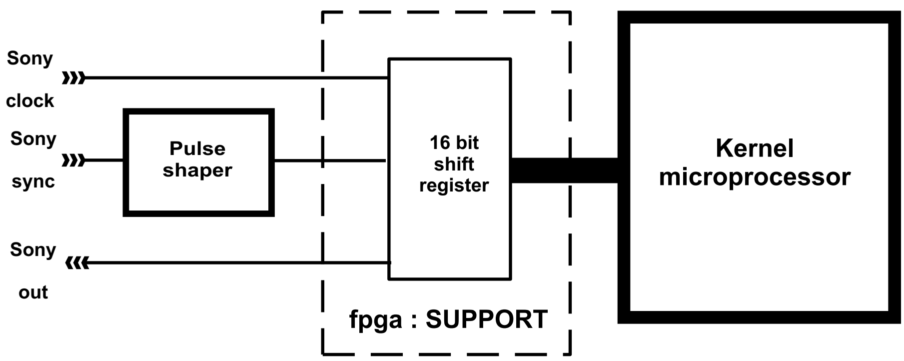

Figure 12 : Sony serial interface.

USER INTERFACE

The ES/2 displays information via a set of dedicated LEDs on the program module and switch board 1 and a proprietary 40 character by 2 line display, as shown in *Figure 13*. The LEDs are all driven by U31 which is an octal latch controlled by the KERNEL microprocessor.

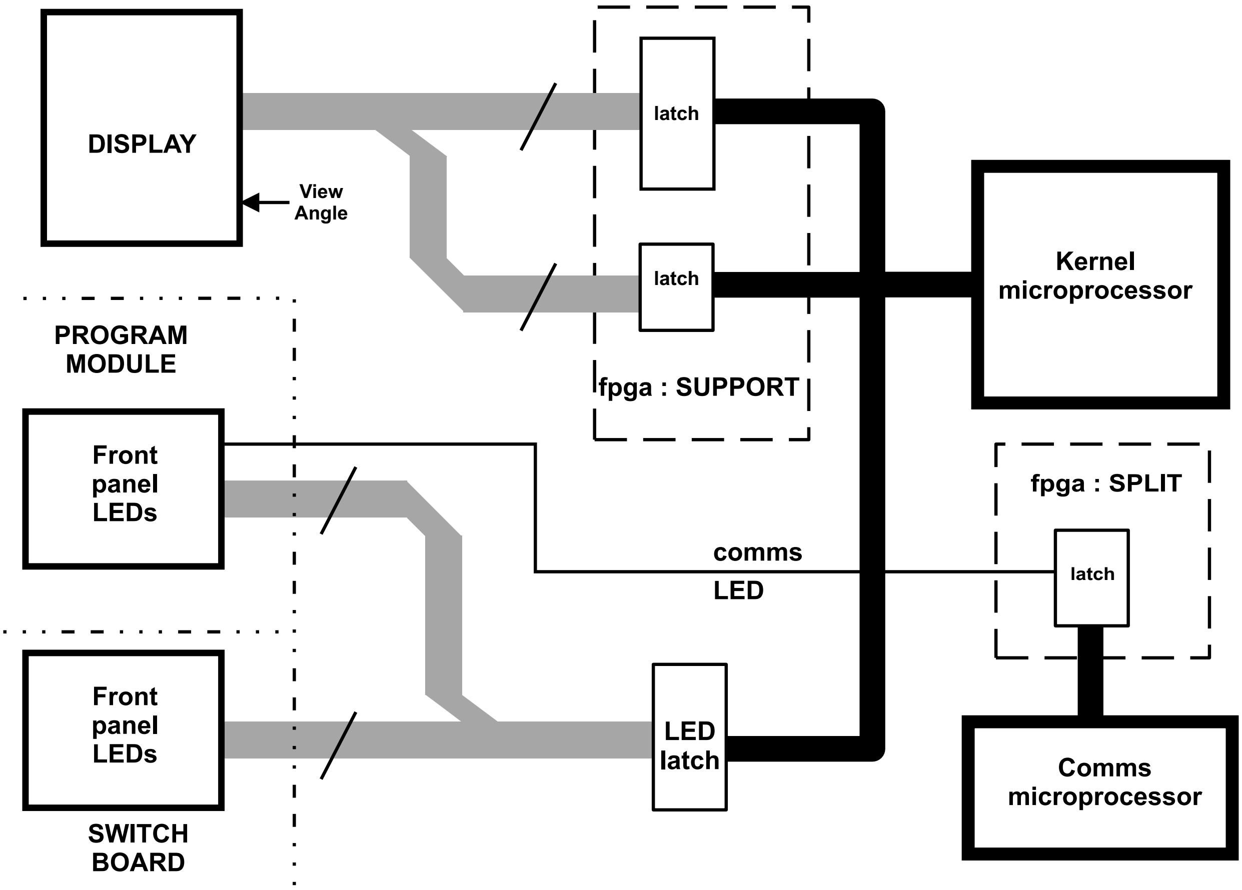

Figure 13 : Display interface

The KERNEL microprocessor controls the display via 3 control lines and 8 bi-directional data lines. The display is not capable of running at the speed required by the microprocessor and so the correct signals are generated by U39, the FPGA SUPPORT. The 3 control signals are :

DISP_RS -	a register select line	Selects between a control and data register.
DISP_EN -	an enable line	The display is enabled when this line is high and the data is clocked on the falling edge.
DISP_RW -	a read / write pulse.	This signal is high when the microprocessor requires to read a byte from the display.

The switch interface, shown in *Figure 14*, has 2 select lines and 8 read lines. One select line, EN_SW, enables the front panel switches and the other, EN_HEX, enables the hex switch and links on the program module.

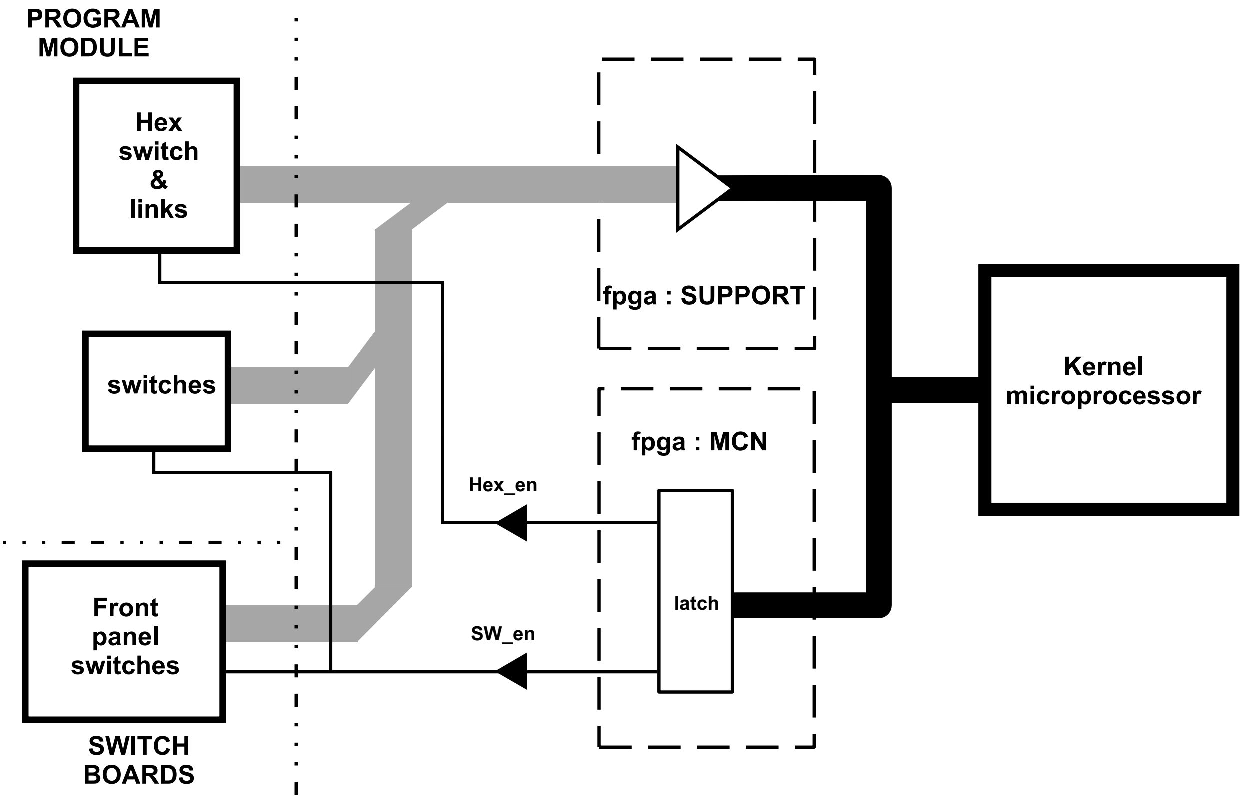

Figure 14 : Switch inputs

The switches are polled approximately every 20ms by the KERNEL microprocessor, the keys are debounced in software.

References

The ES/2 has 7 sources of reference to which the unit can reference and is capable of generating 2 reference signals. There are 3 68B40 timer counter chips (CTCs) which measure/generate these reference signals. *Figure 15* shows a block diagram of the reference signals.

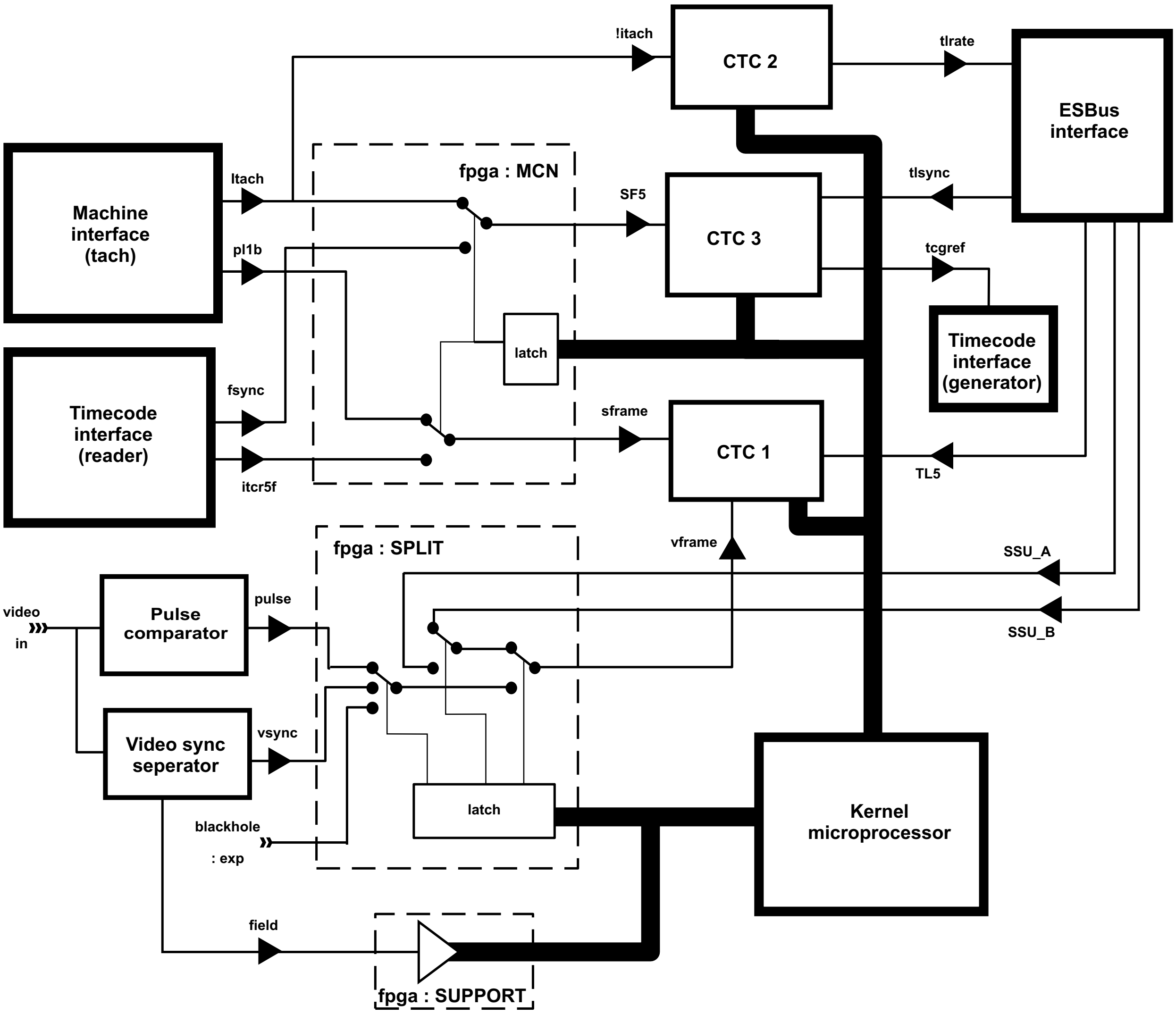

Figure 15 : Reference signals within the ES/2

The reference signals can be split up into 3 groups :

Machine reference signals :

Tach signal.

LTC signal.

The signal SFRAME is used when the ‘MACH’ reference is selected for the timeline or generator reference. The FPGA MCN switches between the frame rate ITACH signal and the LTC frame signal FSYNC depending on whether the machine is a tach or code locked machine. Similarly PL1B is a 5x tach rate signal and LTCR5F is a 5x LTC signal and is switched to generate SF5. This signal is used for servoing the machine.

External reference signals:

Video sync.

Square wave pulse.

Blackhole EXP, for future expansion.

These signals are switched inside FPGA SPLIT to generate the signal ‘VFRAME’ and reflect the VIDEO and PULSE selections for the timeline and generator reference settings.

System reference signal:

Timeline reference used to reference the generator.

Servoing a machine

TLsync is the frame rate system reference, TL5 is a 5x signal phase locked to the timeline. These are the 2 signals used as a reference to servo the machine to. During servoing the ES/2 takes account of the system and machine positions and attempts to adjust the machine’s speed to bring the machine to within 1 frame of the system. Once within 1 frame the signal SF5 is matched to TL5 to bring the machine into close lock. The signals TL5 and SF5 both generate an interrupt and reset a counter inside a CTC. The counter is used to time the phase of the signal.

Generated reference signals

There are two generated reference signals are :

TLrate

This is the timeline signal which can be distributed to the ESbus to provide a system wide reference.

TCGREF

The timecode generator reference.

Both of these signals are generated by a CTC and can wither be synced in software to the signal VFRAME, SFRAME or be free running (XTAL reference). TCGREF can also be synced to the timeline signal TLSYNC. TLSYNC differs from TLRATE in that it is local to an individual ES/2. It can be driven directly by one of the ESbus timeline signals or be a phase locked derivative, used for mixed code. TLRATE however is solely used to drive the ESbus timeline when the RS422 buffers are enable i.e.. the unit is a timeline master on that bus.

KERNEL Counter Timer Chips.

The kernel microprocessor has 3 68B40 counter timer chips (CTCs) which are used to time the various reference input signals and generate the timeline and generator signals. Each CTC has 3 timers which can be configured in a variety of ways.

The following table shows how each of the individual timers in the ES/2 are used :

Timer number		Signal	Use
CTC 1			
	timer 1	TL5	measure the phase of the timeline for servoing
	timer 2	VFRAME	measuring the period and phase of video in reference
	timer 3	SFRAME	measuring phase of the MACH reference
CTC 2			
	timer 1	-	count down timer used for record events etc.
	timer 2	TLSYNC	measuring the phase of the timeline
	timer 3	ITACH	providing a good/bad indication of tach
CTC 3			
	timer 1	SF5	measuring the phase of the machine for servoing
	timer 2	TLSYNC	output a timeline signal for driving the ESbus timeline
	timer 3	TCGREF	output a generator reference signal

Timecode interface

LTC reader

Figure 16 shows a block diagram of the reader.

U6D converts the balanced input to a single ended signal while U6A, B, C and U18 is responsible for converting the analogue signal to a TTL level.

The processor interface is provided by the Audio Kinetics custom LTC reader gate array (U27). This reads the TTL level LTC, separates and sorts he data and interrupts the KERNEL processor every frame edge or every 1/20th frame rate if no code is present. During winding, if a tach signal is present, the processor will turn off the interrupts from the U27 when the machine goes above 8 x play speed and turns them back on when the machine falls below 5 x play speed. This threshold can be altered with the parameter t_spd, see machine parameter description for further details.

The LTC gate array also provides the frame rate signal FSYNC which can be used to lock the generator or timeline signal to MACH reference. The signal !ITCR5F is a 5 x frame rate signal which is used for servoing.

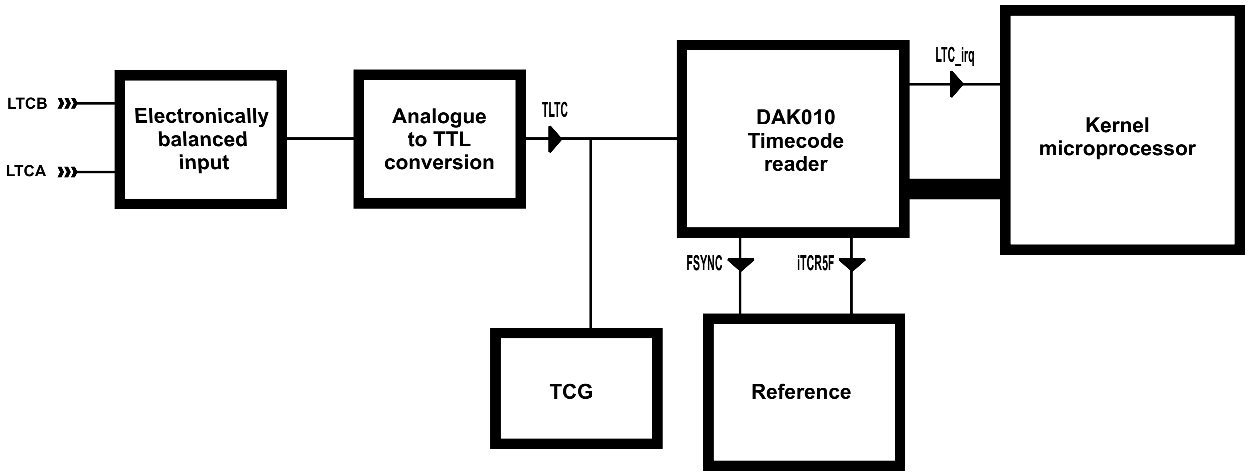

Figure 16 : LTC reader

Timecode generator

The timecode generator can be switched to output 1 of 3 selections :

Generated code under processor control.

Timecode from the ESbus timecode bus A or B.

Reshape the incoming LTC.

Figure 17 shows a block diagram of the timecode generator.

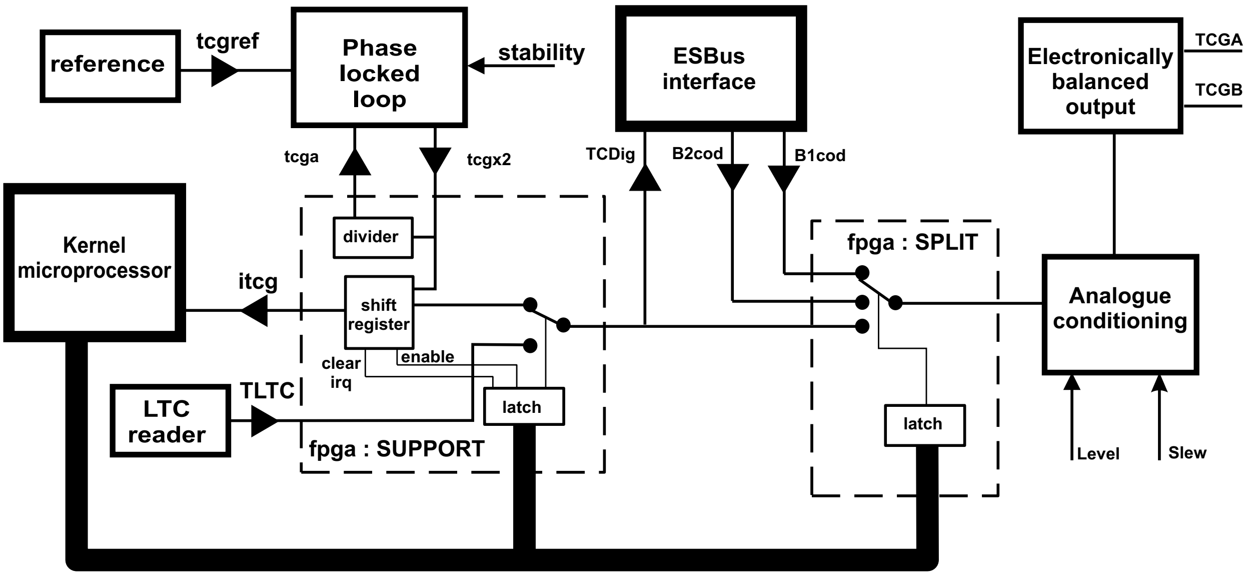

Figure 17 : Timecode Generator

The main generator circuitry is situated inside U39 the FPGA SUPPORT. The generator clock is generated by a phase lock loop U30 locked to the TCGREF signal, which is generated by CTC3. This clock, TCGX2 is used to clock data onto a serial line via a shift register. The shift register interrupts the KERNEL processor every 8 bits which loads the next byte into the register, with 1 level of buffering. The processor can alter the path of the generator signal between the shift register and the LTC reader signal TLTC to generate TCDIG. This signal is fed to the ESbus interface to drive the ESbus timecode bus, if selected as master, and also into FPGA SPLIT - U38 - along with the timecode from both ESbus timecode busses. The processor can select which of these 3 signals drive the generator output.

The TTL generator signal TCRET is conditioned by U7 and converted to a balanced output by U19.

Generator adjustments

VR2 is used to adjust the phase locked loop for maximum stability. VR1 adjusts the slew rate of the generator which should be set to 12.5ms for SMPTE or 50ms for EBU code generation. VR6 on the program module adjusts the output level of the generator.

AK ESbus Interface

The ES/2 communicates with controllers and other ES/2s over 1 of 2 ESbus ports, Bus A or Bus B. Both busses have the following features :

RS422 serial comms ports.

Timeline bus - to distribute a system wide reference square wave.

TLExt line - open collector line to indicate the presence of a timeline master.

Timecode bus - to distribute a system wide timecode stream.

Frame bus - to distribute a system wide video locked square wave

Drec - a Direct Record Line to eliminate serial delays for consistent punch-ins.

Communicating over the ESbus RS422 port is the exclusive job of the COMMS processor. *Figure 18* shows a block diagram of this part of the ES/2. U44 is a 84C42 dual serial control chip which drives U4A to transmit on ESbus A and U4B to transmit of ESbus B. The RTSA and RTSB lines from U44 are used to enable the transmitters onto the ESbus. U4D and U4C can transmit onto the ES/2 receive bus to allow a chase master to send messages to other ES/2s on the bus. Enabling these transmitters is done via the DTRA and DTRB lines from U44. All transmit enable lines are active high.

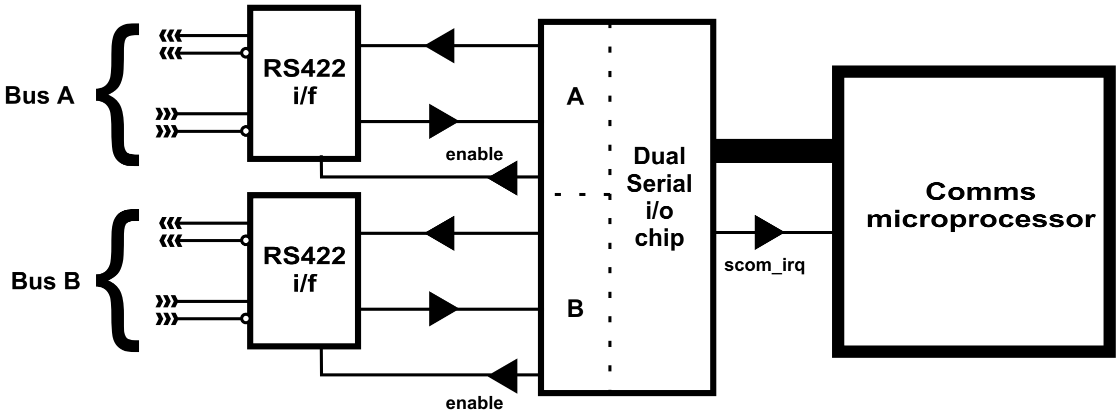

Figure 18 : COMMS processor ESbus interface

The KERNEL processor is responsible for all of the control signals on the ESbus ports. *Figure 19* shows a block diagram of the KERNEL processors' contribution to the ESbus interface.

The timeline bus is driven by the signal TLRATE generated by CTC3 straight into U5C and U5D for ESbus A and B respectively. FPGA MCN outputs the bus enable lines busen1 and busen2. U12A and U12B receive the bus A and B's timeline respectively, the outputs of which are switched by the FPGA MCN to provide the ES/2 with its timeline - TLSYNC.

The TLext line is driven by FPGA SUPPORT through an open collector inverter U17B and U17D for bus A and B respectively. The TLext line is held low if a ES/2 is currently the timeline master i.e. it is driving the timeline bus. The FPGA SUPPORT is able to read the TLext line of both busses to determine if there is a timeline master on the bus. If a unit is requested to become a timeline master when there is already one present then the message "TIMELINE MASTER CONFLICT" will be displayed and the attempt will be aborted.

The Timecode bus can be driven by U16A and U16C for bus A and B by the signal TCDIG from the generator, these are enabled by FPGA SPLIT. U12C and U12D receive the timecode bus of bus A and B respectively.

The frame bus receivers of bus A and B are U11C and U11D respectively. The signal can be used to reference the timeline and timecode generator. Each ES/2 has the capability of driving the frame bus, however, there is currently no facility for activating the drivers in an ESbus system. USA and ABBE are the frame bus drivers of bus A and B respectively.

The ESbus mute lines are generated by open collector inverters U17A and U17C for bus A and B respectively. FPGA SUPPORT can read the state of the ESbus mute line via buffers U29C and U29D for bus A and B respectively.

The drecon lines are active low signals and are buffered by U22B and U22C. It is fed into FPGA SUPPORT which switches through the signal of the selected bus. The sense of the line is converted to generate the KERNEL processor interrupt line idrecon.

Mixed Code Operation - Difflock

In order to operate in mixed code mode the ES/2 references a phase locked loop to the ESbus timeline signal - tl_600 - which is a system standard square wave. The phase locked loop U33 multiplies this signal up to 600Hz (a common multiple of 24, 25 and 30). FPGA MCN divides down the 600 Hz, by a rate controlled by software, to provide a signal which matches the machine standard. FPGA MCN selects the timeline - TLSYNC - between the 2 busses and the difflock output.

U32 generates the 5 x timeline signal - TL5 which is used for servoing. FPGA MCN generates a signal tl_irq which is used to interrupt the KERNEL processor. It is a low going signal on the falling edge of the timeline which is cleared when the processor services the timeline interrupt.

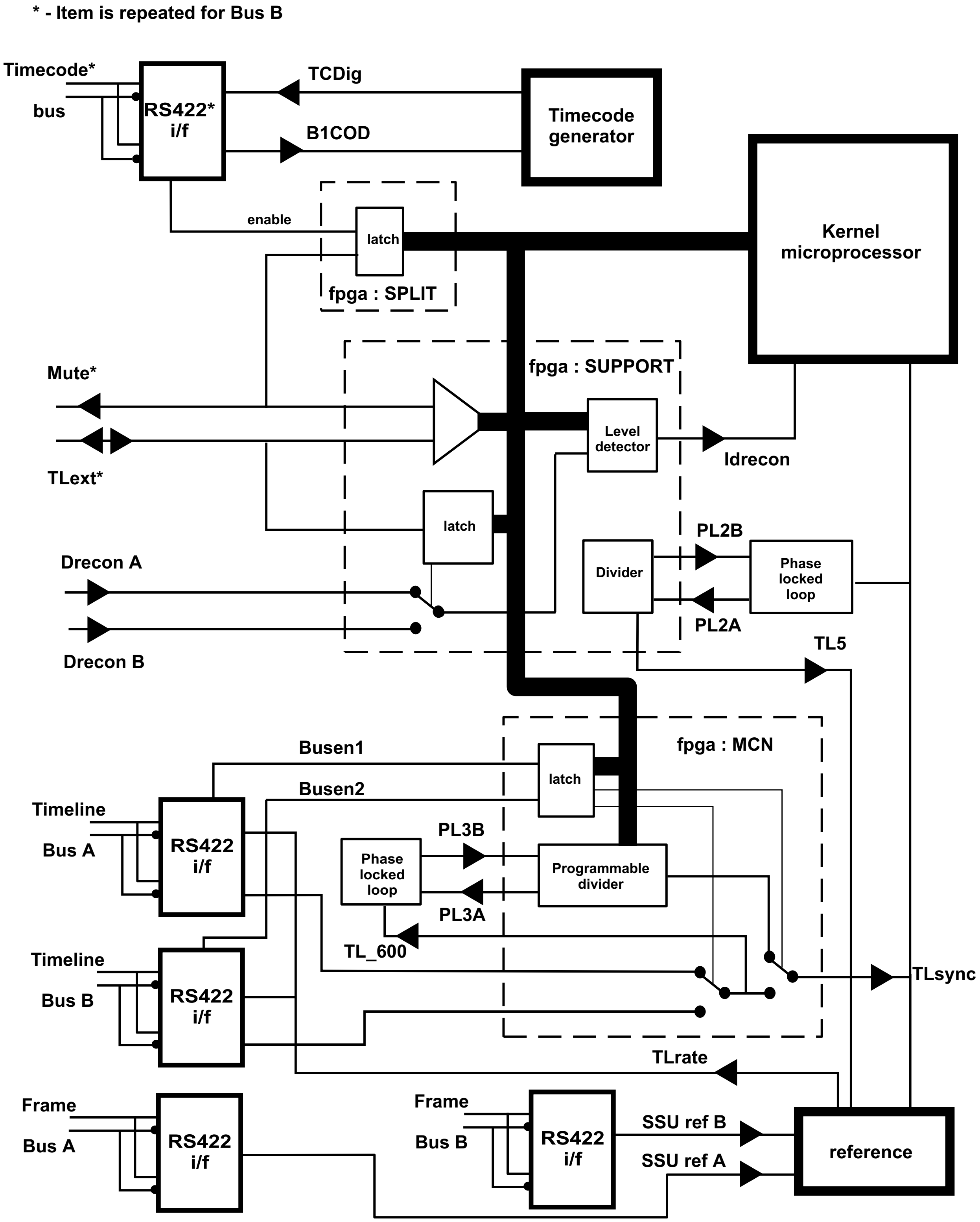

Figure 19 : Kernel processor ESbus interface

ES/2 TEST SOFTWARE

The ES/2 has 30 test routines built into the software. They are called up when the unit is powered up with the hex switch on the program module to 0F16.

On entry the ES/2 checks if there is any RAM available by storing and recalling a two byte number. If the number read was different to that saved then the following message will be displayed and the test routines will halt.

```
RAM cannot be accessed properly
       TEST TERMINATED
```

The ES/2 will then store and recall the numbers AA16 and 5516 to each RAM location between 0000 and 1fff16. If this test fails the test software will display :

```
Bad RAM location
TEST TERMINATED
```

If this test passes then the KERNEL processor will do the following :

A software controlled reset is issued to the peripherals via the !SRES line which is ‘blipped’ low for approximately 80ms.

The LTC chip is initialised.

The command latch is enabled by switching on Q2 via the EN_OP line

The LEDs controlled by the KERNEL are turned off.

The front panel switches are enabled by setting the SW_EN line high.

The ICU chips (U51 and U52) are initialised.

IRQ and FIRQ interrupts are disabled.

The COMMS processor is told to enter the test routines. The state of the comms LED reflects the current state of the COMMS software :

one flash	Processor is waiting to put a character into the FIFO i.e. FIFO is full.
two flashes	Processor is waiting to receive a character from the FIFO, i.e. FIFO is empty.
three flashes	Processor is running in the main loop.

The COMMS processor spends the vast majority of it’s time waiting for instructions from the KERNEL processor. The COMMS LED therefore is flashing twice for most of the time.

Once the initialisation is complete then the following welcome screen is displayed:

```
Welcome to the 1.12 Test Software
                 Press to continue >
```

Pressing key 4 will enter the test software and the test select screen will be displayed.

There is a standard layout for the test routine screens. The middle of the top line displays the action being taken or requested while the extreme left and right text describes the function of key 1 and key 3. The left hand text on the bottom line displays the number of the currently selected test and the centre of the bottom line displays the selected test routines name. The bottom right of the display gives the function of key 4. In the test select screen the OK key (Key 1) runs the currently

selected test. Key 2 is inactive. UP (key 3) and DOWN (key 4) select the test routine. Key 2 is inactive.

```
< OK      Choose test required      UP >
Test 00    Welcome                DOWN >
```

The keys are not active while the test software is being executed.

When a test is completed and passed the user will be asked to press key 4 to continue. If a test fails then pressing key 3 will re-run the test while key 4 will continue with the test routines. An indication of the reason for failure will be displayed using the available space left on the display.

THE TESTS

TEST 00 : Welcome

Running this routine will display the welcome message on the screen.

TEST 01 : CRCs

This routine checks the CRC of page 4 of the KERNEL ROM. With the software running up to this point it would be a bit of a surprise if this test failed however, if the CRC check fails the following will be displayed and the test software will halt:

```
The test EPROM fails the CRC test
         TEST TERMINATED
```

If the KERNEL EPROM passes it’s CRC check then the MACHINE ROM’s CRC is checked. If the ES/2 finds no valid MACHINE ROM pages the the following will be displayed and the test software will halt :

```
Machine EPROM fails the CRC test
        TEST TERMINATED
```

If the ES/2 finds a valid MACHINE ROM page with an incorrect CRC then the following will be displayed and the test software will halt:

```
Machine EPROM faulty or not fitted
         TEST TERMINATED
```

If these tests pass then the following screen is displayed and the user should press key 4 in order to return to the test select screen :

```
                Checking CRCs
Test 01       Okay..  Press to Continue >
```

TEST 02 : keys

This test asks the user to press each key in turn. When the requested key is pressed the ES/2 will ask for the next key in the sequence to be pressed. When all keys have been pressed in sequence and read by the ES/2 the test will have passed and the test select menu is automatically entered. The following screen is displayed at the start of the test :

```
          Press the key marked:
Test 02    Menu   .
```

When the ES/2 detects a key press then it displays the name of that key on the right hand side of the '.'. The ES/2 will remain waiting for the requested key to be read.

TEST 03 : LEDs

This routine displays the following screen:

```
Check the lower row of LEDs are cycling
Test 03                  Press if okay >
```

The ES/2 will light each of its KERNEL LEDs in turn starting on the left had side of the program module with the Bus ON LED and ending on the switch board with the GOOD CODE LED. The LEDs will cycle in this way until key 4 is pressed.

TEST 04 : cmd o/ps.

In order to see the command relays active, the parallel machine connector dongle, detailed in appendix A, is required for this test. This routine displays the following screen:

```
 Check the command outputs are cycling
Test 04                  Press if okay >
```

This test energises each command relay briefly in turn. The associated LED on the command outputs of the parallel machine connector dongle should light in turn. The routine will continue running until key 4 is pressed.

TEST 05 : Tally's

The parallel machine connector dongle, detailed in appendix A, is required for this test. It provides a path from the command relays outputs to the tally inputs. The test first sets the tally polarity by setting the TPOL line high. It then briefly energises each the command relays 0 through 3 and reads the tally input. If the response is not what was expected then the following is displayed and the test fails.

```
    Tally Input   Failed       Rerun >
                        Press to continue >
```

If the first part of the test passes then the TPOL line is inverted and the ES/2 checks that the data read from the tally latch has inverted. If this is not the case the the following is displayed and the test fails.

```
    Tally Polarity Failed      Rerun >
                        Press to continue >
```

If either of these tests fails then pressing the Rerun key (key 3) will restart the test. If the tests pass then the following screen is displayed:

```
              Tally inputs Check
   Test 05      Okay..  Press to continue >
```

After the test has been completed, whether it has passed or failed, the ES/2 will continue energising the command relays and wait for the continue key (key 4) to be pressed. This is to allow some form of debugging to take place.

TEST 06 : CTCs

This test checks the three KERNEL counter timer chips, CTC1 - U54, CTC2 - U55 and CTC3 - U56. Each timer is tested in turn and each test consists of two tests.

1) A check to see if the CTC chip is present. All timers are stopped and preset. The ES/2 reads the CTC and checks that the preset data is read back. If the data is different to that expected then the following message is displayed.

```
    Timer X faulty or             re-run >
    not fitted         Press to continue >
```

2) The second check starts timer 1 going, reads in the timer value after a delay and checks that the counter has actually started. It it has not then the following screen is displayed :

```
    Timer X                       re-run >
    counting error     Press to continue >
```

If either test fails then pressing the re-run key (key 3) will start the test from the point which failed and continue through the tests. If the continue key (key 4) is pressed the the CTC test is exited. If all timers pass the tests then the following screen is displayed and the ES/2 will wait for the continue key (key 4) to be pressed.

```
                 Timers check
   Test 06       Okay.. Press to continue >
```

TEST 07 : film Io

In order to see the film output, the parallel test connector dongle is required for this test. The two film LEDs should be observed. The routine displays the following screen :

```
     Film output LEDs should be showing
   Test 07      bi-phase    Press if okay >
```

The film outputs are set to bi-phase, started and enabled. The servo generator is set to approximately 4Hz. This should result in a bi-phase output of 1Hz. The ES/2 will continue to output the bi-phase until the continue key (key 4) is pressed.

TEST 08 : film hi

In order to see the film output, the parallel test connector dongle is required for this test. The two film LEDs should be observed. The routine displays the following screen :

```
     Bi-phase should be running faster
  Test 08                  Press if okay >
```

The film outputs are set to bi-phase, started and enabled. The servo generator is set to approximately 16Hz. This should result in a bi-phase output of 4Hz. The ES/2 will continue to output the bi-phase until the continue key (key 4) is pressed.

TEST 09 : film stp

In order to see the film output, the parallel test connector dongle is required for this test. The two film LEDs should be observed. The routine displays the following screen :

```
     Bi-phase should be going on and off
  Test 09                  Press if okay >
```

The film outputs are set to bi-phase and enabled. The servo generator is set to approximately 16Hz. This should result in a burst of 4Hz bi-phase output which will stop at each of the bi-phase states in turn. The ES/2 will continue to output this pattern until the continue key (key 4) is pressed.

TEST 10 : film t/d

In order to see the film output, the parallel test connector dongle is required for this test. The two film LEDs should be observed. The routine displays the following screen:

```
     Now output is tach and direction
  Test 10                  Press if okay >
```

The film outputs are set to tach and direction, started and enabled. The servo generator is set to approximately 16Hz. This should result in a tach rate of 8Hz with the direction line changing state every two seconds approximately. The ES/2 will continue to output tach and direction until the continue key (key 4) is pressed.

TEST 11 : FM out

This routine displays the following screen:

```
   Check that the FM out is sweeping with
  Test 11       a scope    Press if okay >
```

The ES/2 sets the servo generator to 400Hz output, starts and enables the film output and sets the output for tach and direction. The servo divider is then used to sweep the frequency of the FM output in the following manner:

The divider is set to two for 2.5 seconds a frequency of 17.9Khz approx is output.

The divider is incremented every 30ms until 7 seconds later it is set to 256.

The divider remains set to 256 for 2.5 seconds outputting a frequency of 137Hz.

This pattern is continually output until the continue key (key 4) is pressed.

TEST 12 : DC out

This routine displays the following screen:

```
DC Servo out is ramping -10 up to +10
Test 12 (may be clipped) Press if okay >
```

The ES/2 sets the servo generator to 400Hz, sets the DC gain to 128 and the initial DC offset to 0. The DC offset is then continually incremented up to 256 and decrements down to 0. This results in a sawtooth wave form with a period of approximately 100ms and a voltage range of approximately ±10V, as shown in *Figure 1*.

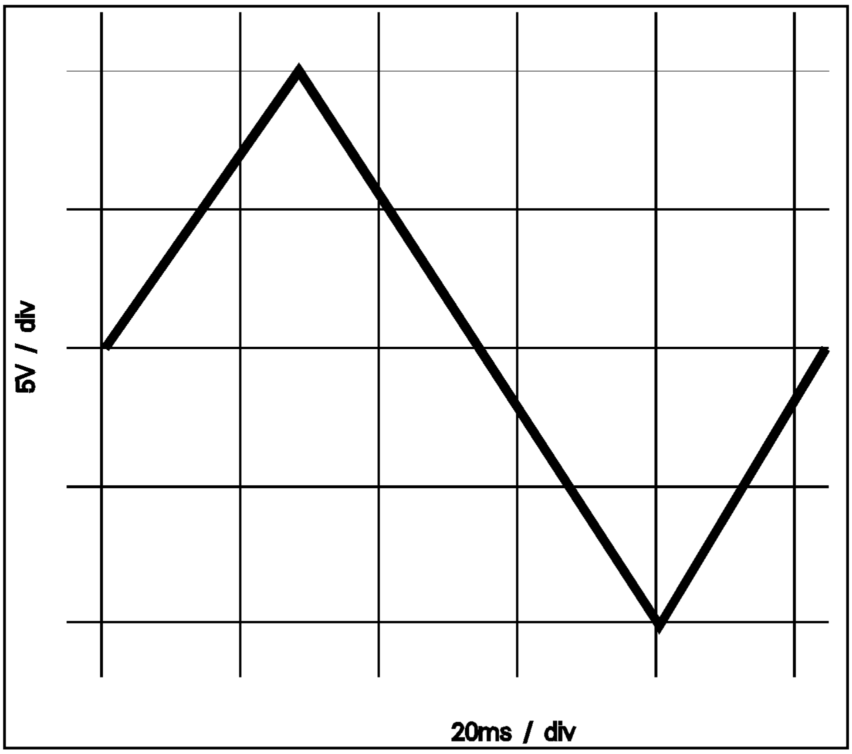

Figure 1 : DCout - Sweeping offset

The pattern is continually output until the continue key (key 4) is pressed.

TEST 13 : servo a

This routine displays the following screen:

```
  DC Servo out is ramping -6 up to +6
 Test 13 (May be clipped) Press if okay >
```

The ES/2 sets the DC offset to 128 which should set the DCout signal to operate around 0V. The initial DC gain is set to 256 and the servo generator is set to an initial frequency of 400Hz. The servo generator frequency is then continually swept up to a frequency of 3.24kHz and the down to 245Hz. This results in a form of distorted sine wave with a period of 266ms and a voltage range of ±6V as shown in *Figure 2*.

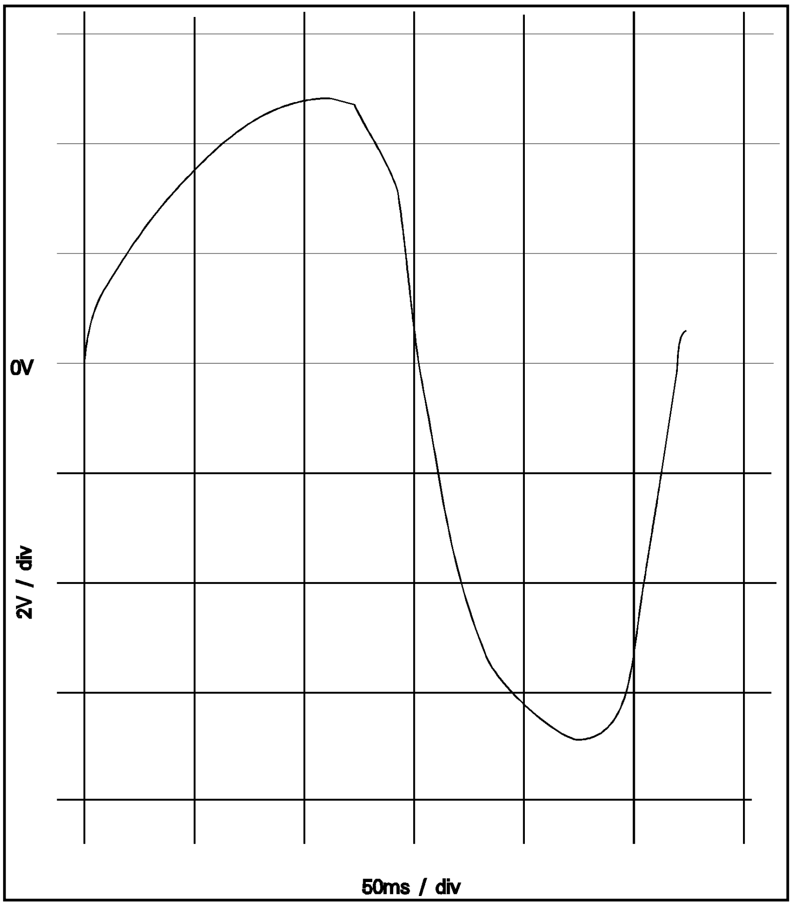

Figure 2 : servo a - sweeping frequency

TEST 14 : DC out b

There are two halves to this test. The first displays the following screen :

```
DC Servo out is ramping 0 down to -10
Test 14 (May be clipped) Press if okay >
```

The ES/2 sets the DC offset to 128, to set the DCout to operate around 0v, with an initial gain value of zero. The servo generator is set to a value of 800Hz. The DC gain in then continually swept from 0 up to 256 and then back to 0. This results in a sawtooth wave form with a period of 105ms and a voltage range of 0 to -7V as shown in *Figure 3*.

This pattern is continually output until the okay key (key 4) is pressed when the second half of the test is entered. This test will display the following screen:

```
DC servo out is ramping 0 up to +5V
                      Press if okay >
```

The ES/2 changes the servo generator to output a frequency of 200Hz. The gain is then cycled round in the same way as above. The results are a sawtooth wave form still with a period of 105ms but now with a frequency range of 0 to +5V as shown in *Figure 4*.

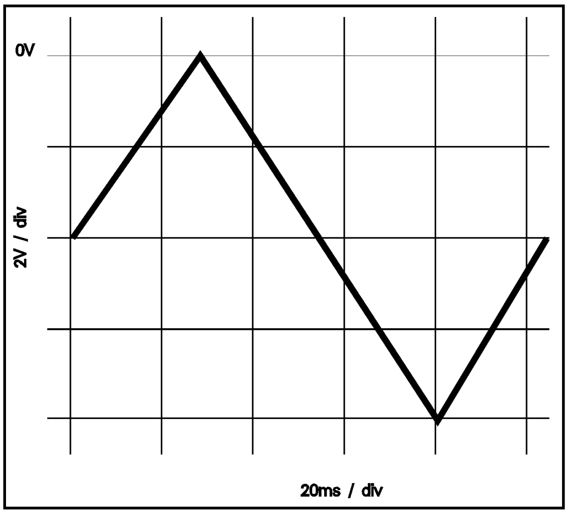

Figure 3 : dc out b-sweeping gain f=800Hz

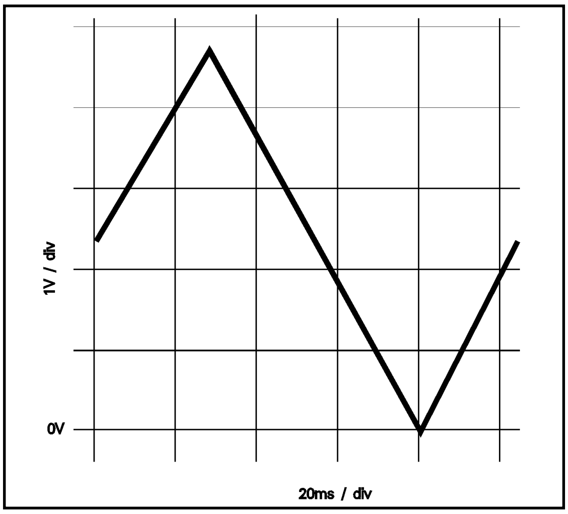

Figure 4 : dc out b-sweeping gain,f=200Hz

TEST 15 : mach sel

This test requires the parallel machine connector dongle in order to read the resistors on the cable selector input. The ES/2 sets the DC gain to 112 and reads the state of the MS1 and MS2 lines via the tally latch. If these lines are not both low then the test fails. If they are both low then the DC gain is set to a value of 120 and the state of MS1 and MS2 is checked. If these lines are now both high then the test passes and following screen is displayed:

```
      Checking machine cable selector
Test 15      Okay..  Press to continue >
```

If the test fails then the screen displays:

```
Machine Cable Selector          re-run >
circuitry failed      Press to continue >
```

In either case the DC gain is continually incremented until the continue key (key 4) is pressed to allow the user to investigate the circuit further.

TEST 16 : tach dir

The parallel machine connector dongle is required for this test, in order to loop the FILMA and FILMB outputs into the TACHA and TACHB inputs. The test dongle allows the ES/2 to switch the TACHA signal between SONY_SER output and FILMA output using the servo relay. The servo relay has to be switched on (connecting the SERVO_COM to the SERVO_NO) for this test to work.

This routine sets the film output and the tach input to bi-phase. It then toggles the film output's direction and checks that the tach direction read changes. If the tach direction does not match the film direction the the following screen is displayed and the user is asked to re-run the test (key 3) or continue with the next test (key 4).

```
  Tach dir out to in            re-run >
 failed (Bi-phase)   Press to continue >
```

The next test sets the film output and tach input to tach and direction. It again toggles the film direction output and displays the following screen if the tach input does not match what was expected.

```
  Tach dir out to in            re-run >
 failed (Tach+Dir)   Press to continue >
```

The software will then wait until the continue key (key 4) is pressed to enter the test select menu or the re-run key (key 3) to restart the test from the beginning.

If the test passes the screen will show :

```
  Checking Tach dir out to Tach dir in
Test 16      Okay..  Press to continue >
```

TEST 17 : tach spd

This test requires the parallel machine connector dongle in order to loop the FILMA and FILMB outputs to the TACHA and TACHB inputs. The routine energises the servo relay to switch the FILMA onto TACHA.

The ES/2 sets the servo generator to 400Hz, sets the tach and film circuits to operate with tach and direction then enables and starts the film outputs. The tach divider is disabled and the tach X2 function is enabled. The following screen is displayed:

```
                Checking Tach input speed
        Test 17
```

The microprocessor Interrupts are then enabled for a specific time and the number of tach interrupts is counted. If the number of tach interrupts is outside the range expected then the following is displayed :

```
      Tach bad at ITACH             re-run >
      (via TachX2)       Press to continue >
```

The software then waits until either the re-run key (key 3) is pressed, in which case it re-runs the tach X2 check, or the continue key (key 4) is pressed in which case it continues with the tach speed checking.

If that test passes (or when the continue key is pressed) the tach X2 function is disabled and interrupts are again enabled for a specific time. If the the number of tach pulses counted differs from that expected then the ES/2 displays the following screen:

```
      Tach bad at ITACH             re-run >
                         Press to continue >
```

Again the ES/2 waits for the user to press either the re-run key (key 3),when the second test is repeated, r the continue key (key 4) which returns to the test select menu.

If the above tests pass then the following screen is displayed:

```
            Checking Tach input speed
    Test 17      Okay..  Press to Continue >
```

TEST 18 : tach div

This test requires the parallel machine connector dongle in order to loop the FILMA and FILMB outputs to the TACHA and TACHB inputs. The routine energises the servo relay to switch the FILMA onto TACHA.

This test sets the tach and film hardware to tach and direction signals, enables the tach divider and sets the divide rate to 16. It sets the film direction to forwards and enables processor interrupts for a given time. If the number of tach interrupts is not in the range expected then the following screen is displayed:

```
      Tach divider output wrong     re-run >
                         Press to continue >
```

The ES/2 will wait until the user either presses the re-run key (key 3) when it re-runs the divider test or the continue key (key 4) in which case it continues with the next divider test.

The ES/2 will then set the film direction to backwards and repeat the above test. If this test fails then the following screen is displayed :

```
   Tach divider output wrong     re-run >
   (reverse)          Press to continue >
```

Pressing the re-run key (key 3) now will re-run the whole tach divider test, pressing the continue key (key 4) will enter the test select menu. If the test passes then the screen will display:

```
      Checking Tach divider accuracy
 Test 18       Okay.. Press to continue >
```

TEST 19 : setup

This test has 4 parts to it and allows the test engineer to set up the tach and servo phase locked loops. It also prompt the test engineer to record the value of DCout when DC gain and DC offset are set to standard values.

On running this test the ES/2 will display the following screen:

```
    Setup the servo phase locked loop
 Test 19              Press to continue >
```

The 1.12 sets the servo generator to 400Hz and waits until the okay key (key 4) is pressed. The test engineer can then set up the servo phase locked loop for minimum jitter.

Pressing the continue key (key 4) allows the engineer to set up the 5x tach phase locked loop. The parallel machine connector dongle is required for this test. The ES/2 sets the film output to tach and direction, enables and starts the film output. The tach circuitry is set to tach and direction, and the divider is enabled and set to 7. Tach interrupts are enabled and the ES/2 waits for the user to press the continue key displaying the following:

```
    Setup the 5x tach phase lock loop
                      Press to continue >
```

When the user presses the continue key (key 4) the ES/2 disables tach interrupts, sets the DC gain to 0 and the DC offset to 128. The DCout signal should be 0V plus errors due to component tolerances. The screen shows:

```
   Check the DC offset
                      Press to continue >
```

When the user presses the continue key (key 4) the ES/2 sets the DC gain to 128 and the offset to 0. The DC out should be close to the -12V rail. The following screen is displayed:

```
   Check the DC gain
                      Press to continue >
```

Pressing the continue key returns to the test select screen.

TEST 20 : timeline

The ES/2 runs four checks for this test and displays the following screen:

```
      Checking timeline input and diflock
 Test 20
```

The first check enables the timeline interrupt via CTC3, enables the timeline mastership and counts the number of timeline interrupts received during 1 second.

If the ES/2 does not receive the correct number of interrupts then the following is displayed:

```
Timeline speed bad              re-run >
Test 20              Press to continue >
```

Pressing the re-run key (key 3) runs the first test again while pressing the continue key (key 4) will continue with the test.

The next test enables the diflock to the highest frequency and then the lowest frequency and counts the number of timeline interrupts. If the number is not within the range expected in 1 second then the following is displayed:

```
Timeline speed bad              re-run >
(via diflock)        Press to continue >
```

Pressing the re-run key (key 3) will run the whole test again. Pressing the continue (key 4) will return to the test select menu.

If the test passes then the 1.12 will display the following and wait for the continue key to be pressed.

```
   Checking Timeline input and diflock
Test 20       Okay.. Press to continue >
```

TEST 21 : m/c serl

This test checks the Sony serial circuitry for the 5000 series of video machines, not the sony P2 port. It requires the test machine connector dongle. This test uses the film output to drive the sony_clk input and the command relay 0 to drive the sony_syn input. The sony_out is looped back into the tach input to enable the ES/2 to read the information this line. The servo relay has to be switch off to route the filmA signal to the sony_clk line.

The ES/2 displays the following screen during this test:

```
           Checking Sony serial output
  Test 21
```

The ES/2 sets the film outputs to tach and direction, enabled and started. The servo generator is set to 800Hz and the data pattern 555516 is loaded into the Sony serial shift register. The tach circuitry is set to tach and direction and the divider an X2 function disabled. The processor interrupts are enabled for a given time and the number of tach interrupts is counted. If the number was out of range then the following screen is displayed:

```
  Sony serial out failed        re-run >
  with data $55      Press to continue >
```

Pressing the re-run key (key 3) will run the test again while pressing the continue key (key 4) will run the next part of the test.

For the next part of the test the ES/2 will change the data to f0f016 and re-run the test. If this part of the test fails then the following is displayed:

```
   Sony serial out failed        re-run >
   with data $f0      Press to continue >
```

If the test passes then the following screen will be displayed and pressing the continue key (key 4) will return to the test select menu.

```
         Checking Sony serial output
Test 21        Okay.. Press to continue >
```

COMMS TEST ROUTINES

Tests 22, 23 and 24 are all associated with the comms processor. As discussed at the start of the test software section, the comms processor is requested to enter its test routines when the test software initialises. The comms LED should flash twice to indicate that it is waiting for a command from the fifo.

There are 4 standard error screens associated with the comms/kernel processor interaction rather than the tests themselves. In these error screens pressing the re-run key (key 3) attempts to re-run the previous test and pressing the continue key (key 4) will continue with the test or return to the test select menu as appropriate.

1) If the kernel has waited for a message from the comms which has not arrived then the following screen will be displayed:

```
Timeout waiting for              re-run >
COMMS response         Press to continue >
```

2) If the kernel processor received data from the comms which did not match anything it was expecting then the following screen will be displayed:

```
Bad data received                re-run >
from COMMS             Press to continue >
```

3) If the Comms processor received data from the kernel which did not match anything it was expecting then the following screen will be displayed:

```
Bad data was received            re-run >
by   COMMS             Press to continue >
```

4) If the Kernel processor attempts to send some data to the comms and fails because the fifo reports it is full then the following screen is displayed:

```
FIFO1 is reporting it            re-run >
is full                Press to continue >
```

TEST 22 : coms ram

The kernel will send a message to the comms requesting the comms to check its own RAM. The comms will then write the pattern aa5516 to every 16-bit location of the comms RAM between 0 and 1f0016. The kernel will display the following screen while the comms processor is testing:

```
Comms processor checking its own memory
Test 22
```

If the comms processor does not read the correct pattern from it's RAM then a message is passed to the kernel processor and the following message is displayed:

```
Comms processor RAM failed      re-run >
                          Press to continue >
```

If the test passes then the comms sends the OK to the kernel which displays the following screen:

```
Comms processor checking its own memory
Test 22       Okay.. Press to continue >
```

TEST 23 : comms CTC

For this test the kernel sends a message to the comms requesting it to check it's CTC and then displays the following:

```
Comms processor checking its CTC
Test 23
```

The comms processor initialises the SIO, to prevent spurious interrupts and checks if the CTC can be read by searching for a recognised pattern from the CTC. If it fails to read the expected pattern then a message is sent to the kernel, the test is terminated and the kernel displays :

```
Comms processor CTC             re-run >
faulty/not fitted    Press to continue >
```

If this test passes then all three timers inside the CTC are started and then checked to see if they are counting. This is done by waiting for a reading the counter values after a period of time and checking that the value read is different from the one initialised. If this test fails then a message is passed to the kernel and the test is terminated. The kernel will display the following screen:

```
Comms processor CTC             re-run >
counting error       Press to continue >
```

If the test fails on either of the above 2 points then pressing the re-run key will start the test from the beginning, pressing the continue key (key 4) will enter the test select screen. If the test passes then the kernel displays:

```
Comms processor checking its CTC
Test 23       Okay.. Press to continue >
```

TEST 24 : comms SIO

This test requires the the 15way Bus A and Bus B ESbus connector dongles, detailed in Appendix A, in order to loop the serial transmit back out to the serial receive. The kernel will send a request to the comms processor to check its SIO and display the following screen:

```
Comms processor checking ESbus SIO
Test 24
```

The comms processor then initialises both channels of the SIO chip for 38.4kBaud, 8 data bits, 1 stop bit, even parity, DTR and RTS inactive and the chip's interrupt is enabled. The receive buffer in the comms RAM is cleared and the processor interrupts are enabled. Three characters are transmitted on the A bus and the receive buffer is checked to make sure the same three characters were received. If the characters did not contain the correct three characters then a message is passed to the kernel and the test is terminated. The kernel displays the following screen and pressing the re-run key (key 3) will start the test from the beginning while pressing the continue key will enter the test select screen.

```
Comms SIO faulty                re-run >
or ESbus A bad         Press to continue >
```

If the Bus A test passes then the test is repeated on Bus B. If this fails then the following screen is displayed:

```
Comms SIO faulty                re-run >
or ESbus B bad         Press to continue >
```

If both busses pass the test then the following is displayed:

```
Comms processor checking ESbus SIO
Test 24       Okay.. Press to continue >
```

TEST 25 : ram msgs

For this test the kernel tells the comms to wait for a specific time while the kernel fills up its fifo with 512 bytes. The following is displayed while this is happening:

```
Sending messages to comms processor
Test 25
```

When 512 bytes have been written to the fifo the kernel will read the fifo's full flag and if it does not register full then the following is displayed.

```
FIFO1 does not report           re-run >
full                   Press to continue >
```

After the timeout the comms processor will empty the fifo.

Pressing the re-run key (key 3) will start the test from the beginning while pressing the continue key (key 4) will enter the test select menu and tell the comms to exit this test.

If the fifo does register full the the following message will be displayed while the kernel waits for a specified time. The Comms checks that the correct data was read from the fifo. The following will be displayed during this part of the test:

```
      Waiting for comms to interpret data

```

If the data read from the fifo by the comms at this point then the kernel displays the message : Re-run will start the test again from the beginning while continue will enter the test select menu.

```
Bad data received               re-run >
from COMMS           Press to continue >
```

If the data read by the comms was OK then the comms fills up its fifo with data and the kernel reads and checks the data. The kernel displays the following screen during this part of the test:

```
Reading data from comms fifo

```

If the comms has filled up its fifo and the fifo does not register full then the kernel is told and the following is displayed:

```
FIFO2 does not report           re-run >
full                 Press to continue >
```

If the data from the comms checks out then the following screen is displayed and the kernel waits for the user to press the continue key (key 4) before displaying the test select menu.

```
Reading data from comms fifo
              Okay.. Press to continue >
```

TEST 26 : mute/tl

This test requires the the 15-way Bus A and Bus B ESbus connector dongles, detailed in Appendix A which 'diode ORs' the tlext and mute lines into the drecon line. The routine selects bus A, sets the mute and tlext lines low and enables the drecon interrupt. The following screen is displayed while this test is running:

```
     Checking TlExt and Mute lines
Test 26
```

The processor interrupts are enabled and the tlext line is toggled high and then low and the processors interrupt is disabled. If no drecon interrupt was received then the processor will display the following and wait for the user to press the re-run key (key 3), when the test will be repeated, or the continue key (key 4) to run the mute test.

```
     TlExt on Bus A faulty      re-run >
Test 26              Press to continue >
```

The mute test follows the same pattern as the tlext test. If no drecon interrupt is received when the mute line is toggled then the following is displayed:

```
     Mute on Bus A faulty       re-run >
Test 26              Press to continue >
```

Both tests are performed on bus B, if the bus B's tlext line fails then this screen will be displayed:

```
      TlExt on Bus B faulty      re-run >
 Test 26                 Press to continue >
```

or if bus B's mute line fails then this screen will be displayed:

```
      Mute on Bus B faulty       re-run >
 Test 26                 Press to continue >
```

If both busses pass the test then the following is displayed and the continue key must be pressed to enter the test select screen.

```
      Checking TlExt and Mute lines
 Test 26                 Press to continue >
```

TEST 27 : serl m/c

This test requires three dongles which loop back the various transmit lines into the receive lines on the machine and diagnostics port detailed in Appendix A. The diagnostics dongle tests the diagnostics port while the RS422 serial dongle and the RS232 serial dongle tests the two standards available on the serial machine control port.

The first part of this test initialises both ports to 38.4kBaud, 8 data bits, even parity and 1 stop bit. DTR and RTS are set inactive and the SIO interrupts are enabled. The following screen is displayed while this test is running:

```
        Checking machine control SIO
  Test 27
```

The processors interrupts are enabled and an ASCII string is transmitted on the serial machine control port. After a delay the interrupts are disabled and the receive buffer checked for characters. If the ASCII string was not correctly received then the following screen will be displayed:

```
 Machine control SIO            re-run >
 faulty or missing   Press to continue >
```

Pressing the re-run key (key 3) will repeat the serial control port test while pressing the continue key (key 4) will run the diagnostics port test. The diagnostics port test follows the same pattern as the serial control port test. The screen displayed while this test is running is as follows:

```
       Checking diagnostics SIO
 Test 27
```

If this test fails then the display will show:

```
 Diagnostic SIO                 re-run >
 faulty              Press to continue >
```

Pressing the re-run key (key 3) will repeat the diagnostics test while pressing the continue key (key 4) will return to the test select menu. If the test passes then the following is displayed and the ES/2 will wait for the continue key (key 4) to be pressed before returning to the test select menu. The re-run key (key 3) will run repeat the whole test.

```
kernel SIO OK                  re-run >
                      Press to continue >
```

As the ES/2 has both RS422 and RS232 standards available on the serial control port, the test should be repeated with the alternative test dongle plugged into the control port.

TEST 28 : frme i/p

This test requires the the parallel machine connector dongle, detailed in Appendix A, in order to provide a signal, generated by the film circuitry, for the sync inputs to detect. The user has to connect the sync-in BNC test plug to the ES/2 video in socket. The display shows the following screen during this test:

```
 Checking frame/sync input and busses
Test 28
```

The 1.12 sets up the film interface to tach and direction, started and enabled. The servo relay is turned off. The servo generator is set to 100Hz in order to generate 50Hz on the FILMA output. CTC interrupts are enabled and timer2 of CTC1 is reset. The hardware is set up to receive its reference from the pulse decoder. The processor's interrupts are enabled for a specific time and the number of interrupts is counted. If an incorrect number of interrupts were counted then the following screen is displayed:

```
Frame signal not reaching      re-run >
CTC1 (direct)        Press to continue >
```

The signal is routed to the ESbus 'frame bus' of bus A and the test is repeated. If an incorrect number of interrupts are received then the following screen is displayed.

```
Frame signal not reaching      re-run >
CTC1 (via bus 1)     Press to continue >
```

Pressing the re-run key will repeat the above test while pressing the continue key (key 4) will continue with the next test.

The signal is then routed to the ESbus 'frame bus' of bus B and the test is repeated. If an incorrect number of interrupts are received then the following screen is displayed:

```
Frame signal not reaching      re-run >
CTC1 (via bus 2)     Press to continue >
```

Pressing the re-run key will repeat the above test while pressing the continue key (key 4) will continue return to the test select menu. If all tests passed then the following screen is displayed and the user should press the continue key (key 4) to display the test select menu.

```
 Checking frame/sync input and busses
Test 28       Okay.. Press to continue >
```

TEST 29 : firq off

This test simply enables the processors FIRQ interrupts for a given time and counts the number of interrupts received with is the time. During the test the ES/2 displays :

```
               Checking FIRQ line
    Test 29
```

If the number of interrupts was below the number expected then this screen is displayed:

```
     FIRQ frequency too low!        re-run >
                         Press to continue >
```

If too many interrupts were received then this screen is displayed:

```
     FIRQ frequency too high!       re-run >
                         Press to continue >
```

In both cases pressing the re-run key will repeat the test while pressing the continue key (key 4) will return to the test select menu. If the correct number of interrupts were received then the screen will show:

```
               Checking FIRQ line
    Test 29       Okay.. Press to continue >
```

TEST 30 : EEPROM

This test checks the EEPROM on the program module for functionality. It fills up the EEPROM with data and attempts to read it back. While the test is running the following screen will be displayed:

```
               Checking EEPROM
    Test 30
```

If the data read does not match the data written then the screen shows:

```
    EEPROM failed to save/restore   re-run >
    data                 Press to continue >
```

Pressing the re-run key (key 3) will repeat the test again. The continue key (key 4) will return to the test select menu. If all is OK with the EEPROM then the screen shows:

```
               Checking EEPROM
    Test 30              Press to continue >
```

Pressing the continue key (key 4) returns to the test select menu.

MACHINE PARAMETERS

Introduction

This document applies to Kernel ROM v1.08 and machine ROM v5.04.

The ES/2 machine interface may be configured to work with a wide range of Audio, Video and Film as well as virtual machines. Optimum performance may be obtained from widely different machines by careful adjustment of a set of machine parameters, accessible via the ES/2 display. The Machine Rom contains all the sets of machine parameters currently provided by Audio Kinetics as well as some software routines for providing machine specific commands. In operation the machine parameters modify the behaviour of the Kernel Rom which performs all the generic functionality of the ES/2.

This document is intended to describe these parameters and give some details of their use but it is assumed that the reader is familiar with current tape machine technology and with interfacing requirements.

The parameters are described in the order they appear in the ES/2 display editor. Gaps have been left between logically grouped parameters to allow for future expansion. Only six characters are used in the ES/2 display so where this differs from the full software label the display name is given at the right of the page.

The parameters are organised into 12 groups as follows :

SERVO PARAMETERS

TACH PARAMETERS

PLAY AND WIND PARAMETERS

TEST PARAMETERS

LOCATE AND CHASE PARAMETERS

TALLY PARAMETERS

COMMAND PARAMETERS

COMMAND ROUTINE POINTERS

UNIFILM PARAMETERS

SERVO CONTROLLED WIND

SERIAL COMMUNICATIONS CONTROL

MACHINE GENERAL DATA

Access to Parameters

To access the Machine Parameter Menu

➤ Press the MENU key to display the Menu Select Menu from the Normal Display.

➤ Press the MACH softkey

The Machine Menu will be displayed.

➤ Press the PROG softkey.

The Machine Parameter Menu will be displayed.

✍ Access to the Machine Parameter Menu will not be allowed if parameter protection is ON.

To turn off parameter protection

➤ Press the MENU key to display the Menu Select Menu from the Normal Display.

➤ Press the LOCAL softkey.

The Local Setup Menu will be displayed.

➤ Press the OPT softkey.

The Local Options Menu will be displayed.

➤ Press the STEP softkey until the Parameter protection option is displayed.

➤ Press the OnOff softkey to turn Parameter protection off.

The message next to the LOAD softkey will change from ‘ok’ to ‘Press’.

➤ Press the LOAD softkey.

The change will be saved and message next to the LOAD softkey will change back to ‘ok’.

➤ Repeatedly press the MENU key until the display returns to the Menu Select Menu or the Normal display as required.

Or

➤ Press the MODE key to return directly to the Normal Display.

The Machine Parameter Menu displays a four digit hexadecimal address with an ‘=’ followed by the two digit hexadecimal value stored at this address. This is followed by a full stop and the name of the parameter.

To select a machine parameter

➤ Press the DIGIT scan softkey to position the cursor on the last digit of the address.

➤ Press the + or - softkeys to change that digit to the desired number.

➤ Press the DIGIT scan softkey to move the cursor to the third digit of the address.

➤ Press the + or - softkeys to change that digit to the desired number.

The parameter name after the full stop should display the required name.

✍ The first two digits of the address cannot be changed. They are fixed as ‘10’.

To change the value of a machine parameter

- ➤ Press the DIGIT scan softkey to position the cursor on the digit that you wish to change on the right hand side of the ‘=’ sign.
- ➤ Press the + and - softkeys to increment or decrement the digit.

The message next to the LOAD softkey will change from ‘ok’ to ‘press’.

- ➤ Press DIGIT scan again if you wish to change the other digit, and use + and - to change the value.
- ➤ Press the LOAD softkey.

The new value will be saved and the LOAD message will change back to ‘ok’.

Resetting Default Parameters

To restore the original machine parameters

- ➤ Access the Machine Menu.
- ➤ Press the LOAD softkey.
- ➤ The parameters for the displayed machine file will be reloaded from ROM.

SERVO PARAMETERS

The servo system uses a 24-bit number to set the output frequency of a programmable divider.

There are three possible servo outputs for a machine, these are FILM, FM and DC. For a given SERVO VALUE, the frequency of the FILM output is:

$$\frac{8000000}{VALUE}$$

For a tach+dir output, or half this for a bi phase output. More about FILM output later.

The frequency of the FM output is calculated as:

$$\frac{\frac{8000000}{VALUE} * 96}{SVODIV}$$

Thus with SVODIV set to 2 and NOMPLY = $9C40 we get the usual 9600Hz FM servo output.

SLOLIM $1000/1/2

This 24 bit number determines the slowest speed at which the ES/2 will attempt to make the machine run in external speed control. The number is a measure of the servo period and so the larger it is the slower the machine will run.

NOMPLY $1003/4/5

This 24-bit number determines the nominal play speed for machine under external speed control. It may be modified by the kernel to reflect the speed at which the machine last locked to the timeline.

FSTLIM $1006/7/8

This 24-bit number determines the fastest speed at which the ES/2 will attempt to make the machine run in external speed control.

SVODIV $1009

The nominal FM output may be divided by any integer between 1 and 256 using this parameter. This allows many of the more usual FM servo frequencies to be generated while maintaining the resolution in the max/min servo parameters. For example with a value of $9C40 in NOMPLY we can generate the following servo frequencies:

MACHINE	SERVO Frequency	SVODIV
MCI	19.2 kHz	1
MTR90	9.6 kHz	2
STUDER B67	3.2 kHz	6

Other servo frequencies may be generated by changing the values of NOMPLY, SLOLIM and FSTLIM with SVODIV selected to keep the resolution in the servo count as high a value as is practical.

The third servo output type is the DC output. A DC voltage of 0 to +5v is generated from a nominally 9600Hz servo. This voltage may be given a range and offset as determined by the following GAIN and OFFSET parameters.

GAIN $100a

This parameter sets the range of the DC servo output up to a maximum of 10v. The ES/2 software recognises that dc servo is in use by a non-zero value in this parameter.

OFFSET $100b

This parameter introduces an offset into the DC servo output such that $00 gives -10v, $80 gives 0v, and $FF gives +10v.

Appendix C suggests a method for adjusting the Offset parameter for a release servo machine.

The TEST parameters (1040 - 104f) contain useful methods for manually experimenting with values for SLOLIM, NOMPLY, FSTLIM, GAIN and OFFSET.

NEGSVO $100c

This parameter allows inversion of the DC servo sense. If it is zero then the DC servo is increased for increasing speed. If the parameter is non zero then the DC servo is decreased for increasing speed.

REVSVO $100d

This parameter is set to a non zero value if the machine speed can be controlled while playing backwards.

RESERVED $100e

MAXSTP $100f/10

The required servo value is calculated 5 times each frame of timecode. This parameter sets the maximum amount by which the servo may be adjusted each time, thus limiting the slew rate of the servo output. This is important if the machine speed is to track the servo output accurately.

This parameter is not used on serially controlled machines.

VIDEO $1011

Video machines are not held in external speed control but are pulled into lock and then released to lock to external video syncs. If this parameter is zero then the machine is always held in external speed control to hold lock. If it is non zero then the machine is pulled into lock, held there for VIDEO servo accesses and then the servo is released.

See REPRATE (1015) for an explanation of servo accesses.

VIDJMP $1012

When a video machine is released from external speed control to lock to external video syncs the tape may jump. In order to achieve a stable lock the ES/2 must pull the machine's position to a point that is the correct number of frames from lock and then release the servo. The machine should then jump the required amount into lock.

Those machines that do not have a repeatable jump characteristic cannot be used with the ES/2 system as video slaves.

NOSVO $1013

This parameter should be set to a non zero value ($FF) if the machine cannot be controlled in external speed mode. This parameter engages the use of Real Master mode in Eclipse or Remote Master mode in Penta, if this mode is possible in the system.

NOWILD $1014

This parameter should be set to a non zero value ($FF) if the machine should be held in external speed control mode for ALL play conditions.

REPRATE $1015 display : REPRAT

It was stated above that the servo output is updated 5 times per frame. Some machines cannot keep up with this rate of update so it may be slowed down by reprate. If this parameter is set to be non zero then each servo access will be followed by REPRATE accesses being missed out, e.g. if REPRATE is 4 then the servo will be updated once per frame (do 1 miss out 4).

DFACTOR $1016 display : DFACTO

This parameter is set for the machines deceleration characteristics when in external speed control. The larger this number is then the earlier the servo will decelerate the machine as it approaches lock. This parameter should be set as small as possible consistent with minimum overshoot, it is only for the case where the machine is lagging and therefore decelerating into lock (see below).

DFLEAD $1017

It has been found that the characteristics of machines differ under deceleration and acceleration so this parameter provides the same function as DFACTOR but for when the machine is leading and needs to accelerate up to lock.

FINEMIN $1018/9 display : FINEMI

When the servo has attained lock using the parameter MAXSTEP the amount by which the servo is changed each update is reduced to the value of FINEMIN. This parameter should be as small as possible consistent with holding lock. The parameter is measured in the same units as NOMPLY. A good machine should hold lock with FINEMIN set to 1 thus allowing a flutter value of less than 0.003% added. This parameter is not used on serially controlled machines.

VHOLDOFF $101a display : VHOLDO

In the case when we have released a video machine from external speed control, its position may be jumping around a bit while it settles down to the external video syncs. This parameter should be set to hold off any servo difference checking while this settling takes place. It is measured in servo accesses (see REPRATE).

If this parameter is set to its maximum value of $FF, the servo is released and not re engaged until the machine has been stopped and started again, this allows the user to examine the difference display to see how far the machine jumps when released.

LETGO $101b

This parameter is the number of subframes away from lock that a video or release digital audio machine will release to external reference. If set to zero then the default of 25 subframes for a video machine and 00 for a digital audio machine will be used. The units are subframes expressed as a hex number.

TAKE $101c

This parameter is the number of subframes that video or release digital audio machine, when released, will be allowed to drift out before the ES/2 servo is re engaged. If set to zero then the default of 50 subframes will be used. If set to $FF then the servo will never re take whatever the error. The units are subframes expressed as a hex number. When this parameter is set to zero and the machine has successfully released, then the retake window widens gradually to 1 frame. This caters for tapes with badly striped timecode.

LOCKAC $101d

This parameter determines number of subframes away from zero difference that the ES/2 is considered to be locked.

SVOJMP $101e

This parameter adjusts the aim point of a servoing machine by one frame. It allows the machine to always pull into lock in the same direction when released to external syncs. It is only used for a release machine controlled by serial only (no LTC).

$00

Always aim to issue a VAROFF command in the exact frame that matches the timeline.

$01

Aim to issue a VAROFF command when the machine is one frame behind the timeline. if approaching the lock point from in front of the timeline (leading).

$ff

Aim to issue a VAROFF command when the machine is one frame in front of the timeline, if approaching the lock point from behind the timeline (lagging).

MAXRATE 101f **display : MXRATE**

When the servo has attained lock and the servo stepsize has reduced to FINEMIN the internal copy of REPRATE above will be increased to the value of MAXRATE to achieve a more gentle servo action. If set to zero then REPRATE is not affected.

TACH PARAMETERS

TACH $1020

This is the number of tach pulses per second at the default speed of the machine, after any programmed divide rate has been applied. It will be updated by the kernel software when a Calibrate is performed. A film machine will use this value to determine its equivalent timecode standard (Film “rate”).

This parameter should be set to zero if no tach signal is present.

TCHDIR $1021

When reading the direction line of the machine this flag allows an inversion to be applied before the kernel decides on the direction of tape motion. If the direction line is high for forwards and low for backwards then this flag should be $FF, if the direction is vice versa then this flag should be zero.

Calibrate updates this parameter.

TCHTYP $1022

If this flag is zero then the hardware is set for a tach+dir signal, if non zero then it is set for a bi-phase tach. When a bi-phase tach is in use the phases should be connected however the user feels and let calibrate sort out the correct value for TCHDIR above.

This parameter is updated by calibrate.

TCHFRM $1023

Some machines generate their tach signal from a control track striped on the tape (e.g. video machines) in which case it is nearly always frame based and as accurate as timecode can be. If this is the case then this flag may be set to the number of tach pulses per frame. The ES/2 will set its tach divider to this number giving an accurate frame rate signal which can replace timecode as a lock source.

TCHDIV $1024

This parameter determines the division rate of the tach hardware and at present will be set to a power of 2 by the calibrate function. The software will aim to get a play rate tach of below 70Hz. Division rates of 0 or 1 will cause the divider to be bypassed.

This parameter is updated by the Calibrate function.

If the most significant bit of this byte is set to 1 then calibrate will not change the divide rate selected but will count the tach rate at the given division ratio. The maximum division rate is $1F (31 decimal).

✍ When calibrate is performed the parameters are updated with the measured integer tach rate but internally an accurate representation of the tach/frame relationship is stored.

CODESPD $1025

This flag allows for machines where there is no tach signal and the frame speed of the timecode is not a true representation of the machines speed, e.g. the Fostex D20.

If this flag is set to non zero then the machine speed is calculated from the rate of change of the Selected Tape Code position. This is not quite as good as normal tach or timecode speed measuring but proves adequate for such obscure cases. The change in position is measured every 8 frames of the timeline.

On a machine where serial position data is available, this parameter is used to prevent ltc being used except when the machine is known to be running at play speed. This overcomes situations where machines output “stationary” or bursts of code while stopped and winding

ACCADJ $1026

When a machine with a resolving servo is used as a chase master, the ES/2 detects a play condition by checking the machine speed and acceleration. It then issues a VARON command. If the machine is accelerating slowly this can cause the servo to be incorrectly engaged. The normal interval between acceleration checks is 200ms. This time can be increased in 50ms steps by ACCADJ.

RESERVED $1027 $102f

PLAY AND WIND PARAMETERS

STARTUP $1030 **display : STARTU**

This is the number of frames taken for the machine to stabilise at play speed from stopped. It is used to hold off the servo until we are sure the machine is running correctly, it therefore is directly responsible for part of the lock time and as such should be set as small as possible consistent with good servo performance.

STOPW $1031

This is the number of frames taken for the machine to stop from play. It is used in locates and chase to achieve accurate parking. Each time the machine stops from play this parameter is adjusted by the kernel to achieve a more accurate park the next time.

MAXWND $1032/3

This parameter determines the maximum allowed wind speed when toggling. It is usually set to a value that is greater than the maximum the machine can achieve but may be set lower by the user. The format of this value is “internal speed format” (ISF) which is a 16 bit number where $0008 represents play speed and multiples are corresponding multiples of play speed.

MINWND $1034/5

This parameter represents the minimum speed at which reasonable toggle performance may be maintained. The kernel will not attempt to control the wind speed of the machine below this value. It is in ISF units.

TOGWND $1036/7

During a chase to playing master with the machine approaching from behind, it must toggle ahead and then stop to wait for the correct play point. When the machine gets within the toggle window (TOGGW) it will use this speed to toggle ahead of the master reference. It is in ISF units.

CWRAMP $1038/9

This parameter determines the slope of the deceleration curve for locate/chase performance. The larger this number is then the steeper the deceleration curve is and consequently the machine braking is applied later. This parameter should be set as large as possible consistent with minimal overshoot.

TOGERR $103a/b

When the controlled wind is toggling the machine to maintain a fixed speed this parameter determines an error window around the required speed such that if the machines speed is within this window the toggle will be equal between the two wind commands to maintain speed. Only if the machine speed goes outside of this window will one or other of the wind commands be missed out to allow acceleration/deceleration. The parameter is in ISFormat for ease of use by the kernel software but it is likely to be in the range of about $0004 to $0008. TOGERR should not be set higher than MINWND.

Some machines respond better if the toggled commands are a wind command (determined by required direction) and the stop command. If this flag is set to non zero then the deceleration command will be a stop command.

TOGTYPE $103c display : TOGTYP

This number determines the type of wind speed control applied to the machine; when toggling

0	normal wind command toggling.
1	DC speed control (from DC servo output).
2	FM speed control (from FILMA output).
3	Control of machine speed by Serial Comms.
4	alternative wind command toggling.
5	normal wind command toggling with no DEADZ used
6	toggling using FM speed control and SONY 16-bit Serial

Speed control types 1,2 and 3 are controlled by further parameters, see later.

TSTOPW $103e

This parameter serves the same purpose as STOPW but is measured from the machine toggling at PRKWND speed. It is updated by the kernel in the same way as STOPW.

RESERVED $103f

TEST PARAMETERS

TESTIT $1040

This parameter allows an experienced ES/2 user to perform experimental writes to parts of the ES/2 hardware.

When this parameter is set to $48 (ASCII H) you may write directly to the command output, servo control latch and servo dividers. Parameters $1041 to $1045 are the values written to the hardware.

Alternatively with this parameter set to $53 (ASCII S) the diagnostic status screen will be enabled under the front panel "mode" button.

T_SERV $1041/$1043

This three byte value is the value poked into the 24-bit servo counter.

T_GAIN $1044

This value is written to the gain control port of the DC wind/servo hardware.

T_OFFSET $1045

This value is written to the offset control port of the DC wind/servo hardware.

T_CMD $1046

This value is written to the command output of the ES/2 parallel machine port. See the machine control board schematic.

bit 0 .... pin 1 play
bit 1 .... pin 2 stop
bit 2 .... pin 3 record
bit 3 .... pin 4 forward wind
bit 4 .... pin 5 rewind
bit 5 .... pin 6 pause / edit
bit 6 .... pin 7 rehearse
bit 7 .... pin 8 unlace / lift defeat

T_CONT $1047

This value is written to the servo control latch.

bit 0 .... svopol - servo polarity
bit 1 .... film stop
bit 2 .... fdir - film direction
bit 3 .... mode - film output enable
bit 4 .... tc bus enable 1
bit 5 .... servo relay
bit 6 .... tally polarity
bit 7 .... tc bus enable 2

T_COMS $1048

Byte passed to Comms for debug purposes.

bit 0 Cancel emulator increments of time by 1 frame.
bit 1 ESbus time outs disabled.
bit 2 Set alternate bus chase master (Comms software revision 2.04 - 3.00 only).
bit 3 Send EOL to Zeta controller.
bit 4 Allow chase slave to follow external unit with same address.
bit 5 Send timeline tick to comms when timeline master.
bit 6 Sony emulator sends tl runs to kernel (do not use chase master routines).
bit 7 Emulator does not run timeline after a delay when play command received.

T_IFUP $1049

Update an Information field to the comms.

T_LTC $104a

Number of frames of good consecutive monotonic values of ltc required to establish “Good” code after a drop out. If this parameter is set to 0 then a default value of 3 is used.

T_GEN $104b

Number of incoming erroneous frames before a jamming generator will re jam. If this parameter is set to 0 then a default value of 5 is used.

T_SYS $104c

Manual setting for “system” standard.

01 = 24 fps
02 = 25 fps
03 = 30 fps
13 = drop frame

A hard reset should be performed to clear the effects of using this parameter.

Setting bit 7 of this parameter will prevent the ltc reader from updating the machine standard. Using bit 7 for this purpose does not require a reset to clear its effects.

T_SPD $104d/4e

As machine speed increases, provided that there is a tach signal available, the ES/2 ltc reader is switched off to avoid processor overloading. This parameter can be used to override the default speed at which this takes place (5x play speed). Units are ISF.

T_SYNR $104f

Synchronisation tests.

bit 0 Pulse the unlace/lift defeat relay on timeline second boundary.
bit 1 Pulse the rehearse relay on machine second boundary.

bit 2 Move Evertz/SSL serial timecode back by 2 frames (Screensound version 4).

LOCATE AND CHASE PARAMETERS

TOGGW $1050

During locate or chase functions the last stage may be a toggle into park or a toggle ahead of playing master reference, this parameter represents in binary frames the size of the window within which fixed speed toggling takes place. This parameter should be set large enough to accommodate the toggle stopping window (TSTOPW).

PLAYW $1051

During locate or chase functions the last stage may be a play into park, this parameter represents in binary frames the size of the play into park window.

PRKERR $1052

This parameter, in binary frames, represents the allowed error in parking the machine. If the machine stops outside this window it will attempt to repark. It may attempt to repark up to PRKJOG (105d) times before abandoning the job, it will then leave the machine where it stands.

STRTFRM $1053 display : STRTFR

Usually the machine is parked ahead of the timeline position and when the timeline starts to run this value represents, in signed binary frames, the correct difference between machine and timeline to issue the play command to the machine. It is adjusted by the kernel on each play to achieve a faster lock time next time. A value of 5 frames means the play command should be issued when the timeline is equal to the machine position 5 frames.

Positive values may be used, whereupon the play command is not issued until after the timeline has passed the machines position. This can compensate for those machines which start with a boost, causing them to travel at faster than play speed for a brief period before settling down.

PRK_AHD $1054 display : PRK_AH

This is the number of frames a chasing machine will stop ahead of a stopped master reference. It should be large enough to accommodate STRTFRM described above but not so large as to delay the synchronous play of the system.

WNDLAGF $1055 display : WNDLGF

When the machine is chasing, this parameter defines how much to lag behind a forward moving master. The algorithm to define the number of seconds to follow behind the master is:

$$\frac{WNDLAG}{8} \quad \frac{WNDLAG}{16} * masterspeed$$

where master speed is in multiples of play speed

WNDLAGR $1056 display : WNDLGR

Same as WNDLAGF but for a reverse moving master. When setting the wind lag factors there is a worst case condition for each. For the forward moving system the worst case is a master which stops instantly so that the slave has to wind down to a controlled stop without going too far past the stopped master position. For a backward moving system the worst case is for a master to drop instantly into play, the slave must decelerate to stop ahead of the master ready for a synchronous

play command. Any overshoot in either of these cases causes an untidy and unacceptable performance. These parameters are best set in conjunction with CWRAMP described earlier.

ULTIME $1057/8 **display : ULTIME**

When a video machine is chasing or locating it will attempt to unlace the tape and perform an unlaced wind if the difference to be made up is greater than this value. The ability to lace or unlace the tape is reflected in the command bytes ‘LACE’ $108d and ‘UNLACE’ $108e. This value is in BCD minutes and seconds (i.e. only use digits in the range 0-9), minutes in $1057 and seconds in $1058.

LTIME $1059/a **display : LTIME**

When a video machine is locating or chasing and the tape has previously been unlaced as described above then it will attempt to reduce the difference below this value before it will lace the tape up again. Conditions apply as for ULTIME, it is also BCD minutes and seconds (i.e. only use digits in the range 0-9).

PRKWND $105b/c

During a chase or locate operation when the machine gets within the toggle window it will use this speed to toggle into park ahead of the master reference. It is in IS Format. If this parameter is set to 0 then the value of MINWND will be used for this operation.

PRKJOG $105d

Maximum number of attempts to get within PRKERR (1052) of the locate target. When parameter is set to 0 the default value of 3 is used.

SVOWIN $105e

This sets the number of frames away from lock that the machine must be before the servo system causes a chase operation to be started in order to rejoin the timeline. If this parameter is set to zero then a default value of 80 frames (50hex or 3secs 5frames @25fps) is used. Units are binary frames

RESERVED $105f

TALLY PARAMETERS

AVTALS $1060

Each of the top four bits of this byte should be set to 1 if the corresponding tally is connected. The bits are defined as:

bit 4 MACHINE IN RECORD

bit 5 MACHINE RECORD ENABLED

bit 6 END OF TAPE

bit 7 GENERAL USE

TALINV $1061

If a connected tally is active low the corresponding bit in this byte should be set to a one.

FLSHTL $1062

If a connected tally is expected to flash for a required function then the corresponding bit in this byte should be set to $01.

FLSHPRD $1063 display : FLSHPR

If any tallies are expected to flash then this parameter should set the flash period (approximately) measured in 20ms steps.

NEGTAL $1064

The tally interface is usually expected to work from a positive voltage source. If a machine works from a negative voltage then this parameter can be set to non zero to configure the inputs to work in this way.

SERTAL $1065

A value of $01 in this parameter indicates the tallies come across the serial link. A value of $02 does not use the serial record tally. A zero indicates the usual parallel tallies.

EOTACT $1066

This parameter determines the action of the ES/2 when an end of tape tally has been received. Non-zero will force the ES/2 to go OFF BUS. Zero will cancel all processes and stop the machine without taking the synchroniser OFF BUS.

RESERVED $1067

The following parameters determine idealised tallies to report to the kernel. The idealised tallies are defined as follows;

bits 0-3 are unused

bit4 = 1 machine is in record

bit5 = 1 machine is record enabled

bit6 = 1 machine is out of tape

bit7 = 1 general use defined for specific machines

TAL1FLSH $1068 **display : TAL1FL**

This byte should be set to the idealised tally code to be reported when tally 1 (record) is flashing.

TAL1ON $1069

This byte should be set to the idealised tally code to be reported when tally 1 (record) is on solid.

TAL2FLSH $106a **display : TAL2FL**

This byte should be set to the idealised tally code to be reported when tally 2 (record enable) is flashing.

TAL2ON $106b

This byte should be set to the idealised tally code to be reported when tally 2 (record enable) is on solid.

TAL3FLSH $106c **display : TAL3FL**

This byte should be set to the idealised tally code to be reported when tally 3 (end of tape) is flashing.

TAL3ON $106d

This byte should be set to the idealised tally code to be reported when tally 3 (end of tape) is on solid.

TAL4FLSH $106e **display : TAL4FL**

This byte should be set to the idealised tally code to be reported when tally 4 is flashing.

TAL4ON $106f

This byte should be set to the idealised tally code to be reported when tally 4 is on solid.

COMMAND PARAMETERS

AVCMDS $1070

Each of the bits of this byte should be set to 1 if the corresponding command line is connected. The bits are defined as;

bit 0	play command
bit 1	stop command
bit 2	record command
bit 3	forward wind command
bit 4	rewind command
bit 5	pause/edit command
bit 6	rehearse command
bit 7	lifter defeat/unlace command

CONCMD $1071

If a command must be held continuously until the next command is issued then the corresponding bit in this byte should be set to a one.

CMDINV $1072

Commands are usually activated by a momentary closure of the output relay. If a bit is set to 1 in this byte then the command will be changed to a momentary opening of the corresponding command relay. The momentary action may be modified as described above by CONCMD.

CMDTIM $1073

This byte determines how long a command relay closure will last, measured in 20ms steps. Because of the asynchronous nature of command issuing there will always be an inaccuracy of up to 20ms. The value in CMDTIM represents the minimum time that a command will be maintained.

DEADZ $1074

After a command time has finished this parameter determines how much time should elapse before another command may be issued. This is to stop a machine seeing two separate commands so close together as to be interpreted as a single multi press command. Measurement is in 20ms steps.

CYCLE $1075

When toggling the wind commands, using TOGTYP 0, each command is issued for its CMDTIM time. A period is left between the commands which is determined by this parameter in 20ms steps. For some machines this parameter may be reduced to zero, thus speeding up the toggling and giving a smoother wind performance.

RPTCMD $1076

With some machines repeat presses of the same command button give different functions. If these functions are unwanted this flag may be set to a non zero value ($FF) to stop the ES/2 from issuing the same command twice in succession.

SVOPLY $1077

When synchronous play is required from stop the default action is to enable the servo and then issue the play command. If this flag is set to a non zero value then the order is reversed to be play followed by servo enable. If the parameter video is non zero and this parameter set to zero then only the servo enable command is issued.

SONYINV $1078/9 **display : SONYIN**

The Sony serial output is usually high with active low command bits. If an active high command bit is required then the corresponding bit should be set in this word.

EVTADV $107a

This value is a number of frames that is subtracted from an event time to determine when the event is sent. It is used to account for the serial delays on a serially controlled machine.

VIDTRK $107b

A non zero value in this parameter indicates the record track selection is of the type used by video machines. This parameter controls the use of the Assemble Edit option in the Local Options Menu, and the type of track selection screen offered to the user by an Eclipse controller. In a machine controlled by a Sony P2 dialect, this is the number of bytes in the track selection command.

TRKNUM $107c

This number represents the total number of selectable tracks available on the machine. If bit 7 is set then each track will be selected individually when all tracks are enabled, otherwise an 'ALL ENABLED' message will be sent. This is useful for Studer machines. It is used by the record track selection command (RECSEL) on Audio machines.

TOGTIM $107d

Some machines benefit from a faster action when using wind command toggle modes. In this case a reduced command time can be set for toggling only. Units are 20ms steps. If this parameter is set to zero then toggling uses CMDTIM output pulse width.

UNREC_P $107e

It is normally considered safe practise to always copy an unrecord command to the attached machine even if it is not recording. This protects against faulty machines, broken wires etc. When this parameter is set to be non-zero, the synchroniser will not respond to an unrecord command unless a record tally is present.

RESERVED $107f

COMMAND ROUTINE POINTERS.

The ES/2 kernel has a logically defined set of commands which it may issue to the machine. These logical commands are converted to the closest physical command that the machine is capable of performing by the following byte table. Each entry should contain a number that causes a particular command routine to be run. The byte table is as follows:

NULL	$1080	no action/clear command
PLAY	$1081	play command
STOP	$1082	stop command
CRAWLS	$1083	crawl stop
REC	$1084	record in command
UNREC	$1085	record out command
CRAWLR	$1086	crawl rewind command
CRAWLF	$1087	crawl forward command
RPLAY	$1088	reverse play
FWD	$1089	forward wind command
REWIND	$108a	rewind command
TOGG	$108b	assert toggle mode
REHEARSE	$108c	rehearse command display : REHEAR
LACE	$108d	lace/load the tape
UNLACE	$108e	unlace/unload the tape
LFTDFT	$108f	defeat the tape lifters
LFTOFF	$1090	release tape lifters
VARON	$1091	enable varispeed mode
VAROFF	$1092	disable varispeed mode
PAUSE	$1093	pause command
EDIT	$1094	post stop command
*SERVO	$1095	perform the servo function
*RECSEL	$1096	record track selection
*LOCATE	$1097	cue to a given position
*JOGR	$1098	step one frame backwards
*JOGF	$1099	step one frame forwards
*UNREH	$109a	take machine out of rehearse.
S_PLAY	$109b	release machine to external syncs. If this parameter is zero then VAROFF will be used.
*INIT	$109c	This command is sent when a machine file has just been loaded or in a reset sequence.

✍ Commands marked with an ‘*’ are only used by serially controlled machines.

RESERVED $109d $109f

Appendix A is a list of the current parallel command routines and Appendix B is a list of the current serial command routines available. This list will be updated as machine interfaces require new action routines. If a machine cannot perform a function then the relevant table entry should be set to zero.

Many of the functions will be highly machine specific and therefore are likely to be beyond the scope of most users.

UNIFILM PARAMETERS

MODE $10a0

This flag controls the unifilm output drive. If it is non zero then the FILMA FILMB outputs will be bi-phase signals, if it is zero then FILMB will be a direction signal.

RESERVED $10a1

RAMPUP $10a2

This parameter controls the rate of change (increasing frequency) of the film output for tape speeds above play speed. A larger number represents a gentler ramp. A value of $00 means that the film output frequency will change abruptly to the new value.

Note that the film ramp parameters control the change between different operating conditions, e.g. play to fast wind or toggle into park. When a film machine is being locked to the timeline then the usual servo parameters are used to control the film output frequency.

RAMPDWN $10a3 display : RAMPDW

This parameter controls the rate of change (decreasing frequency) of the film output.

RESERVED $10a4/5/6

PRAMPUP $10a7 display : PRAMUP

This parameter controls the rate of change of the film output when changing from stop to play.

PRAMPDWN $10a8 display : PRAMDN

This parameter controls the rate of change of the film output when changing from play to stop.

RAMPTRM $10a9

This parameter controls the acceleration of the film ramps. It specifies the number of frames taken to transit between full ramp rate and fixed speed. If this parameter is set to $00 then a value of 4 is assumed.

NEWRAMP $10aa

If this parameter is set non zero then an alternative film ramp algorithm is invoked. This number is a single ramp factor which covers all four operating conditions of the above ramp factors. This parameter works in the opposite direction to the original ramp factors i.e. larger number = steeper ramp.

RESERVED $10ab $10af

SERVO CONTROLLED WIND

RESERVED $10b0

SWNDMX $10b1/2/3

This 24-bit number determines the value placed in the servo register for maximum wind speed. For a DC servo controlled machine the smallest (fastest) meaningful value is 00 48 00.

SWNDMN $10b4/5/6

This 24-bit number determines the value placed in the servo register for minimum wind speed. For a DC servo controlled machine the largest (slowest) meaningful value is 01 d4 00.

SWNDGN $10b7

This byte determines the value placed in the DC servo gain register. It is only applicable if DC speed control is selected using TOGTYP.

SWNDOFF $10b8

This byte determines the value placed in the DC servo offset register. It is only applicable if DC speed control is selected.

SWNDNEG $10b9 display : SWNDNE

If DC speed control requires decreasing voltage for increasing speed then this flag should be set to non zero, otherwise set to zero gives increasing voltage for increasing speed. It is only applicable if DC speed control is selected.

SWNDSTP $10ba/b display : SWNDST

This 16-bit number determines the change per command step (20ms) when in servo controlled wind.

The current value in the servo wind control register (24 bits) is divided by 2 SWNDSTP number of times to obtain the correction factor for the new value. Thus if SWNDSTP is set to 1 then half the current value is used as the step size. If SWNDSTP is set to 7 then the servo will change by 1/128th of it’s present value each command step.

The DC output voltage can be assessed from the formula

$$DCout \quad 10 \quad (\frac{20}{256} * SWNDOFF \quad (\frac{975}{25600} \quad \frac{39}{2560} * Vf) * SWNDGN)$$

where Vf is derived from the FM servo dividers (i.e. SWNDGN has no effect when Vf = 2.5v, Vf has no effect when SWNDGN = 0)

CRAWLSPD $10bc/d/e display : CRLSPD

This parameter determines the mechanism used when a crawl is commanded.

For a parallel controlled machine it is a period value, for the servo hardware so the larger it is the slower the machine will move. It may not bear comparison to the normal servo parameters such as NOMPLY.

For a serially controlled machine $10bc/d is the speed the ES/2 will crawl the machine and the units are ISF (PLAY SPEED = 0008). $10be is the speed a machine will be jogged and is related the machine's dialect.

DIRTYPE $10bf display : DIRTYP

When DC servo wind is required to be a continuous sweep of control voltage from MAX reverse speed to MAX forward speed then this value should be non zero. In this case SWNDOFF should be adjusted for the zero speed condition. Bit 7 of this parameter controls whether minimum voltage is max forward or max reverse speed.

RESERVED $10c0 $10cf

SERIAL COMMUNICATIONS CONTROL

SERIAL $10d0

If this flag is set to non zero then the serial machine control port is enabled.

BAUD $10d1

This byte is used by the ES/2 to set the baud rate and number of stop bits as follows:

bits 3-0	baud rate
0000	50
0001	109.2
0010	134.58
0011	150
0100	300
0101	600
0110	1200
0111	1800
1000	2400
1001	3600
1010	4800
1011	7200
1100	9600
1101	19200
1110	38400

bit 5	stop bits
0	one stop bit
1	two stop bits

e.g. Sony machines talk at 38.4k baud with one stop bit. This parameter is set to 000011102 or $0e.

FORMAT $10d2

The ES/2 does not use this parameter, the serial port setup is derived from the dialect.

DIALECT $10d3

This value determines the dialect spoken;

$00	Sony
$01	Studer
$02	Ampex
$03	Studer ESbus

TIMEOUT $10d4

This value determines the comms timeout, measured in 5ms steps and is specified by the machine's communications protocol.

JOGTYPE $10d5 display : JOGTYP

This parameter is zero for the standard ES/2 crawl command or non zero if the machine has a jog command implemented. In this case the commands JOGR/JOGF are used.

LOCTYPE $10d6 **display : LOCTYP**

This parameter is zero for the standard ES/2 locate command or non zero if the machine has a locate command implemented. In this case the command LOCATE is used.

01 = standard absolute locate
02 = relative offset locate

T_POSN $10d7

This is the byte sent to the machine when requesting a tach timer position. If this parameter is set to $ff then no requests for this information are transmitted.

L_POSN $10d8

This is the byte sent to the machine when requesting an LTC position. If this parameter is set to $ff then no requests for this information are transmitted.

V_POSN $10d9

This parameter is not used by the ES/2.

SCYCLE $10da

This is the number of 20ms steps in between status requests to the machine. The status information contains tally and locate status. If this parameter is set to $ff then no requests for status information are transmitted.

Care should be taken that values for SCYCLE and PCYCLE (below) are not be set low enough to cause the ES/2 processor to be slowed down in carrying out normal machine control duties.

PCYCLE $10db

This is the number of frames in between position requests to the machine. If this parameter is set to $ff then no requests for position information are transmitted. When a machine is calibrated PCYCLE is set to 1 in order to get position reports every frame. The machine file value is retained and restored whenever the module is reset. This requires a new calibrate.

RESERVED $10dc

BREAK $10dd

This parameter should be non zero if the machine requires a break character to begin communications.

SVOTYP $10de

This number indicates the type of servo used to synchronise the machine.

$00	Normal parallel FM or DC servo.
$01	Serial SERVO command used until released to external reference (VIDEO).
$02	Serial servo command used continuously.
bit 7	No varoff if chase master.

EVTLKO $10df

This is the number of frames that serial commands are suspended before an event. This is to guarantee that the serial link will not be busy when the command need sending.

MACHINE GENERAL DATA

NAME $10e0 to $10f3

This 20 character string is displayed to tell the user what machine he has. It should be set so that the first 12 characters make a complete name for the NORM display. The remaining 8 characters give any extra information for the MACHINE display.

EOT $10f4

This is the end of text byte for the above name string.

RESERVED $10f5

RSPMAP $10f6 $10f7

This parameter is 'ANDed' with the response to a device type request before a search of subfile is made.

RESPON $10f8 $10f9

This parameter gives the response of the machine to a device type request. If parameter sersel is non zero and the response matches this parameter then the subfile is loaded.

SERSEL $10fa

If this byte non zero then a device type request message will be sent to the machine after a reset. If a response can be matched to a subfile then that subfile is loaded.

MCNTYPE $10fb **display : MCNTYP**

This byte contains the type of machine:

1 = Audio Tape Machine
2 = Video Tape Machine
3 = Film Machine
4 = Digital Audio Machine
5 = Digital workstation

The kernel software makes many decisions based on this parameter so it is important to set it correctly.

CSUM $10fc

This byte is reserved for the kernel to keep a checksum of the file while it is held in RAM.

SUBFILE $10fe **display : SUBFIL**

This byte is reserved for the number of any subfile that has been used to modify the main file.

FILE $10ff

This byte contains the file number of the particular machine that is in use. It is this number which is matched to the cable number, read at power up.

APPENDIX A

List of Real Parallel Command Routines in ROM

Parallel Command Description.

Assert means momentary or continuous depending on the setting in ‘CONCMD’. Continuous means continuous regardless of the setting of ‘CONCMD’. S relay is the servo relay available on pins 11, 12 and 13 of the machine connector. S relay on means pins 12 and 13 connect.

S relay off means pins 12 and 11 connect.

0	nullc0	null
1	playc0	play (assert pin 1)
2	stopc0	stop (assert pin 2)
3	rcinc0	record + play (assert pins 1 and 3)
4	rewc0	rewind (assert pin 5)
5	fewc0	forward wind (assert pin 4)
6	lfdoc0	lifter on (continuous, set pin 8)
7	lfdfc0	lifter off (continuous, clear pin 8)
8	varoc0	servo on (S relay on)
9	varfc0	servo off (S relay off)
A	toggc0	toggle (set toggle mode)
B	fewc1	searchJVC (S relay on, assert pins 4 and 6)
C	rewc1	searchJVC (S relay on, assert pins 5 and 6)
D	varoc1	svo on JVC (S relay on, assert pins 4 and 6)
E	varfc1	svo off JVC (S relay off, assert pin 1)
F	rhinc0	rehearse (assert pin 7)
10	playc1	play JVC 8250 (assert pins 1,4,6)
11	varfc2	servo off JVC 8250 (S relay off, assert pins 1,4,6)
12	rplyc0	reverse play (assert pins 1 and 5)
13	playc2	play command SONY 5000 †
14	stopc1	stop command SONY 5000 †
15	varoc2	servo on SONY 5000 †
16	varfc3	servo off SONY 5000 †
17	fewc2	forward laced wind SONY 5850 †
18	rewc2	rewind laced wind SONY 5850 †
19	nullc1	null command + SONY serial †
1A	toggc1	toggle command with SONY pause †
1B	varfc4	TEST VAROFF COMMAND FOR 3324
1C	lacec0	lace command for SONY 5000 †
1D	unlac0	unlacing stop for SONY 5000 †
1E	editc0	pause/edit (assert pin 6)

1F	varoc4	servo enable with dead zone
20	fcrlc0	forward crawl JVC6600
21	rcrlc0	reverse crawl JVC6600
22	rplyc1	reverse play JVC6600
23	fcrlc1	forward crawl SONY 5000 †
24	rcrlc1	reverse crawl SONY 5000 †
25	varoc5	servo on JVC BR6600E (S relay on, assert pin 8)
26	rcinc1	record (assert pin 3)
27	rcotc0	hold play + short stop
28	lfdfc1	assert pause/edit (continuous, pin 6 on)
29	lfdoc1	de assert above (continuous, pin 6 off)
2A	fcrlc2	forward crawl BVU800 (assert pin 7)
2B	rcrlc2	reverse crawl BVU800 (assert pin 7 and 6)
2C	rplyc2	reverse play BVU800 (assert pin 6)
2D	rhinc1	rehearse OTARI (assert pins 1,3 and 7)
2E	rcotc1	record out JVC8250 (assert pin 8)
2F	varoc3	servo on JVC 8250 (S relay on, assert pin 6)
30	stopc2	stop command JVC (S relay off, assert pin 2)
31	rcinc3	play then record for Sondor Libra
32	stopc3	stop command for Sony 5630
33	reca80	record command for the Studer A80
34	stopa80	stop command for the Studer A80
35	fewc1a	JVC search forwards command without using servo relay
36	pl_stp	Unrecord for Lyrec Frida issue Stop+Play together
37	urec124	Unrecord with a short stop pulse.
38	rewc3	assert rewind line (pin 4), output SWNDMX to dc servo, no servo relay
39	fewc3	assert forward line (pin 4), output SWNDMX to dc servo, no servo relay
3A	varo6	output nomply to DC servo, energise servo relay.
3B	fcrl DC	crawl forward.
3C	rcrl DC	crawl rewind.
3D	toggc3	toggle command not using pause line.
3E	lfttog	lifter defeat which toggles
3F	playa80	play command for Studer A80

Commands 40 to bf are reserved for serial commands

C0	ffa80	deactivate PAUSE then assert FF
C1	rewa80	deactivate PAUSE then assert RW
C2	rcinc4	energise SERVO relay then assert RC

C3	rcotc4	release SERVO relay then assert PL
C4	fewc4	issue STOP then DC servo controlled forward wind
C5	rewc4	issue STOP then DC servo controlled rewind
C6	toggc4	issue STOP then DC servo controlled toggle
C7	stopc4	disconnect servo relay then STOP (assert pin 2) after deadz
C8	toggc5	toggle using pause line but not FWD & RWD lines (no servo relay action)
c9	playc3	assert play line twice
ca	rcinc2	record then play + record

Commands marked '†' are complex and designed specifically for the SONY 5000 range of machines. As such they are of little use for other machines. Call the AMS Neve Customer Support Department if you need details of what they do.

APPENDIX B

List of Real Serial Command Routines in ROM

40	serial play
41	serial stop
42	serial record
43	serial unrecord
44	serial forward wind
45	serial rewind
46	serial lace
47	serial unlace
48	serial pause
49	serial null
4A	serial jog forward
4B	serial jog rewind
4C	serial variable on
4D	serial variable off
4E	serial servo
4F	serial rehearse command
50	serial reverse play
51	serial edit
52	serial lifter defeat
53	serial lifter off
54	serial crawl stop
55	serial record track selection 1 byte
56	serial locate jog forward
57	serial locate jog rewind
58	serial initialise absolute locate
59	serial clog timer enable
5A	serial clog timer disable
5B	serial locate 1 frame
5C	serial locate relative offset
5F	serial edit on
5E	serial edit off
5F	serial position request
60	serial step on 1 frame
61	serial step back 1 frame

62	serial servo using ±25% command
63	serial variable on with col field set
64	serial stop with col field reset
65	serial col field reset
66	serial record track selection 2 bytes
67	serial record track selection 8 bytes
68	serial forward wind or variable wind @ maxwnd
69	serial rewind or variable wind @ maxwnd
6A	serial varon with colour field 2
6B	serial unrehearse
6C	serial forward at 5x speed with laced check
6D	serial rewind at 5x speed with laced check
6E	serial pause and unrehearse
6F	serial test for record then stop/play
70	serial Nagra T's play
71	serial Nagra T's varon
72	serial stop + delayed colour reset
73	serial record track selection masking TC channel bit (1 byte)
74	serial edit off with delay
75	serial pause then stop after a delay
76	serial initialise machine command
77	serial record track selection command for VPR6
78	serial servo command for Ampex PR6
79	serial set play speed to 1xplay and issue play command
7A	serial varon command for Ampex VPR6
7B	serial Studer track select
7C	serial rehearse mode on command
7D	serial servo using shuttle command with restricted servo range
7E	serial transmit offset and enable chase
7F	serial enable chase
80	serial shuttle servo

APPENDIX C

Adjusting OFFSET ($100b) on a DC servo video machine

A video machine (or any other type that uses a release servo mechanism) does not use the dc servo when the machine is put into PLAY. This makes it difficult to find the correct value for the offset parameter which can be critical to good lock times.

The following procedure can be adopted.

- ➤ Make a note of the GAIN parameter (100a) and then set it to $00.
- ➤ Make a note of the VAROFF parameter (1092) and set this also to $00.
- ➤ Put the machine ON LINE to a group, or CHASE ON to a video referenced timeline.
- ➤ Play the group or the chase master.
- ➤ Select the ES/2 difference display.
- ➤ Adjust the value of OFFSET until the difference display stops incrementing. This value will be correct for your machine.
- ➤ Replace the GAIN and VAROFF parameters to their original values